Incorporating Differential Privacy Protection to a Basic Recommendation Engine

Ali Inan

Adana Alparslan Turkes Science and Technology University, Faculty of Engineering,
Computer Engineering Department, 01200, Adana, Turkey.
Tel: +90 322 455 0000. e-mail: ainan@atu.edu.tr

ORCID ID: 0000-0002-3149-1565

Research Paper Received: 04.12.2019 Revised: 17.01.2020 Accepted: 18.01.2020

Abstract—Recommendation engines analyze ratings data to suggest individuals new products or services based on their past experiences. However, the set of items that an individual has rated and the ratings on these items are critical for protecting individual privacy. Existing work on the problem focus on overly complicated recommendation engines. In this study, we concentrate on the case of a very simple engine protected with a very strong mechanism. Towards this goal, we incorporate differential privacy to an item-based neighborhood predictor. Empirical analyses over large-scale, real-world rating data indicate the efficiency of our proposed solution. Even at very high levels of protection, the rate of loss in prediction accuracy is below 5%, a reasonable trade-off for privacy protection.

Keywords—Recommendation Engine, Collaborative Filtering, Differential Privacy, Laplace Mechanism

1. Introduction

The widespread use of the Internet has resulted in a major shift towards e-commerce. Most consumers of today decide on the items to purchase and the particular provider of such items only after researching their alternatives over the Internet. This behavior is not limited to items and extends to services as well. The range of items/services available through e-commerce is very wide and range from hotel reservations to car rentals, from books to movies and music, and even the daily needs of their households. There exist websites whose sole purpose is to provide past user experience and ratings over items and services. Consider Booking.com [1] for hotel reservations and The Internet Movie Database [2] for movies.

Internet websites do not collect user ratings and reviews only to increase their hit rates. The ability to understand why an individual does or does not like an item/service can be key to commercial success. Let us visit the two scenarios below for motivation.

Scenario 1: Customer C is reviewing item I on a website. If the website can successfully infer that C will like I, the site may offer C a discount on I to satisfy its customer. As an alternative, if C is not likely to be content with I, then the same site can propose C another similar item I' that C is expected to like. In both situations (whether C is to like or dislike I), C leaves the website with a good experience and will come back for the next purchase.

Scenario 2: Customer C has so far rated items $\{I_1, I_2, \ldots, I_n\}$. Suppose that the website knows of another customer C' that has rated similar items similarly and this customer C' has also liked item I_y where $y \notin [1, n]$. The similarity of past scores of C and C' together with the high rating of C' on I_y suggests that customer C would like the not-yet-experienced item I_y. By offering C the item I_y, the site can help its customer discover new items and increase its sales.

In both scenarios, the system that infers how a customer will rate an item is called a "recommendation engine" [3]. Recommendation engines inspect all ratings of all customers and use these data to estimate how a customer will rate an item that he/she has not rated before.

Recommendation engines take as input a two-dimensional ratings matrix R. The i^{th} row of R contains the ratings of customer C_i . The j^{th} column of R contains the ratings for item I_j . Consequently, $R[i][j]$ will be the rating of C_i over I_j . Typically, $R[i][j] = 0$ indicates that C_i has not rated I_j.

Despite its rather simple structure, ratings matrix R is a potential threat to individual privacy. Consider hotel reservations. Suppose R reveals that customer C has rated hotel H. Having rated H implies having visited the city/country H is located in - which may contain in itself further implications. If H is in Mecca, then it is heavily probable that C is Muslim (the threat of Islamophobia). If H is in Uganda, C might have gotten infected with the Zika virus during the visit.

Similar examples exist for almost any type of purchase. Having read "The Communist Manifest" alongside many other books of the leftist literature could indicate political view. A highly positive rating on the movie "The Passion of the Christ" could indicate religious belief as well as political view (most individuals would tag the rater as a conservative in the U.S.).

All of these examples reveal that the ratings matrix R contains data that are sensitive to individual privacy. Therefore, R has to be protected against privacy leaks and recommendation engines should be strengthened to support privacy-by-design. Otherwise, not only privacy sensitive customers will withhold from providing their ratings over items/services, but also the above mentioned inferences against individual privacy cannot be prevented. Various studies in the literature have attempted to solve the privacy protection problem during recommendation generation [4], [5].

In this study, we discuss how privacy protection can be incorporated to a basic recommendation engine. Our recommender system predicts the rating that a customer C will give to an item I by first locating neighbor items of I. A neighbor of I is another item that has received similar ratings from the same customer as I. Once neighbors are identified, the predictor than takes a weighted average of the votes of C over the neighbors of I.

The protection mechanism we utilize, namely differential privacy, is one of the strongest mechanisms known yet. The basic approach of differential privacy is to conceal all access to the ratings matrix R by allowing only aggregate statistical queries and adding noise to the results of these queries. As explained in [6] by Dwork, added noise ensures that any possible violation despite such protection is essentially unavoidable - even if the disclosed sensitive data were not part of the R.

Primary contributions of our approach are as follows:

- Privacy of R is protected with the Laplace mechanism of differential privacy. Unlike most of existing work, we provide strong and quantifiable protection.
- Our recommendation engine is one of the simplest engines possible. Existing work on (especially differentially private) privacy preserving recommendation systems focus on overly complicated engines.
- We experiment with large-scale, real-world rating data. Our results indicate that strong privacy guarantees can be provided easily and with little compromises upon recommendation accuracy.
- In accordance with the privacy by design principle [7], we promote item-based solutions in contrast to customer-based solutions, and also argue that sibling data sets should be built over rows of R rather than cells of it [8].
- We review a wide range of different solutions on the problem of privacy preserving recommendation generation.

The rest of the paper is organized as follows. We review existing work on privacy preserving recommendation generation in Sec. 2. Then we provide preliminaries on differential privacy and the employed rating prediction method in Sec. 3. Our proposed solution is presented in Sec. 4 and the experimental results are presented in Sec. 5. We conclude in Sec. 6.

2. Literature Review

Privacy of recommender systems has been studied from different perspectives. We will first review studies that show the need for privacy protection, then, in order, the studies that propose utilizing data perturbation and cryptography. In the literature, there is only one study that employs differential privacy protection with the Laplace mechanism [8]. The difference between the proposed approach and this existing study will be detailed below. For a general overview of recommender systems please refer to [3]. Similarly, privacy over recommender systems are surveyed in [4], [5].

Calandrino et al. [9] motivate the need for privacy protection over recommender systems through an attack scenario. The attack relies on basic information about the individuals that give the ratings and presents how the identity of the owner of a specific row of the ratings matrix R can be disclosed. This study shows that the recommendations reveal personal identity and motivates our proposed solution.

Gunes and Polat introduce another attack scenario called the “shilling attack”, where the attacker adds fake profiles in order to prevent accurate recommendations and potentially discover private data about relevant users [10]. Gunes and Polat propose detection methods against these attacks in [11]. Okkalioglu et al. show that if users provide confidential data in an inconsistent manner, then their profiles can be revealed and their rated items alongside the ratings can be reconstructed [12]. These studies motivate the need for a resilient privacy protection mechanism such as our proposed solution.

In [13], Polat and Wu focus on yielding recommendations over distributed data. Their study proposes adding noise to simple building blocks such as scalar multiplication of a customer’s (resp. an item’s) row (resp. column) of ratings. Added noise is white, i.e., has 0 mean and its magnitude can be adjusted by the participants. White noise ensures that the results obtained after privacy protection do not deviate much from the original results. Polat and Wu also study cases where the data is vertically [14] or horizontally [15] distributed and the recommendations are drawn in a hierarchical manner. Okkalioglu et al. [16] test the effectiveness of the major solutions for vertically partitioned rating data under 3 different attack scenarios. Their results indicate that the collaborating parties learn each other's confidential data. In [17], it is claimed that a clustering of users yield private data with up to 70% accuracy. These results indicate that rather than adding white noise empirically, adhering to a strong privacy mechanism such as differential privacy is therefore vital - as we try to achieve in this work.

Yargic and Bilge extend single-criterion randomized collaborative filtering to multi-criteria collaborative filtering [18], where the customers produce multiple ratings for an item/service in a diverse set of dimensions. For hotel reservation, these could be cleanliness, amenities offered etc. We leave this harder problem definition for future work.

Boutet et al. [19] discuss a scenario that is similar to Polat and Wu [13]. Multiple recommender systems want to collaborate in order to increase their prediction accuracy. Their solution is based on sharing item similarity matrices under differential privacy using the exponential mechanism. Li et al. discuss the case of a peer-to-peer solution based on local differential privacy [20]. Unlike these studies, our problem definition assumes a centralized data setting.

Erkin et al. focus on the distributed rating data problem as well [21]. They propose a secure multiparty computation (SMC) solution that rely on cryptographic primitives like secure sum and secure product protocols executed between multiple participants each holding a share of the distributed data. Similar SMC-based solutions have been proposed in [22], [23].

SMC solutions assume that performing all computation over encrypted data ensures that only the result will available to the involved parties after protocol execution. However, such guarantees are only applicable in distributed scenarios and even in such cases, the heavy computational requirements of SMC protocols are prohibitive.

McSherry et al. present the only work that employs global differential privacy in a centralized, non-distributed case [8]. This solution defines the recommendation generation process as a query set and retrieves the "best" recommendations using the exponential mechanism of differential privacy. Consequently, the ordered list of items to be recommended are perturbed as well. In this study, we restrict ourselves to the Laplace mechanism only. Therefore, there is only one source of randomness in our results, which involves rating prediction. We consider the problem of obtaining an ordered list of recommended items to be merely a budgeting issue and leave this problem for future work.

Unlike our scenario, a limited number of studies in the area consider the possible effects of and remedies to an untrusted recommendation engine. The only solution under such constraints appears to be perturbing rating vectors of users before they arrive at the recommender system. Existing work rely on smooth sensitivity under local differential privacy [24], exponential mechanism [25] and local differential privacy [26], [27].

Solutions by Shen and Jin [24], [25] simply relax the requirements of differential privacy to maintain high accuracy predictions. Shin et al. correctly criticize these approaches for they protect only against disclosure of items rated by a user and not the rating values [26]. Shin et al.'s solution is the state-of-the-art in this line of work. We consider the untrusted server scenario to be overly pessimistic and waive the constraint for future work.

Zhang et al. propose another relaxation of differential privacy called "personalized differential privacy" [28]. In this setting, customers reserve a privacy budget to each item they rate. A similar approach is taken by Xuying et al. in [29], where customers mark their ratings as sensitive or non-sensitive. We deem these approaches to be overly impractical for the customers and unclear in terms of its privacy implications.

Guerraoui et al. present an interesting solution that involves replacing rated items of a user with similar other not-rated items [30]. Their approach assumes existence of distance-preserving items, which is unrealistic. Even if such items were found, an abled attacker could possibly invert the replacement protocol [31].

In [32], Ren et al. propose a differentially private solution using neural network based auto-encoders. They discuss two separate solutions that rely on input perturbation (i.e., local differential privacy) and objective function perturbation (i.e., the exponential mechanism).

Based on this review of the literature on preserving individual privacy in recommendation generation, our proposed solution stands out because (i) we rely only and entirely on the Laplace mechanism of global differential privacy, (ii) we show that even very simple algorithms can attain reasonable accuracy under excessive noise. Our work does not suffer from high computational costs as in the case of SMC solutions [21], [22], [23], and provides rigorous privacy guarantees unlike the case of most random perturbation solutions [17].

3. Preliminaries

This section introduces preliminaries on differential privacy in Sec. 3.1. We then discuss a basic, yet very heavily used rating prediction method in Sec. 3.2.

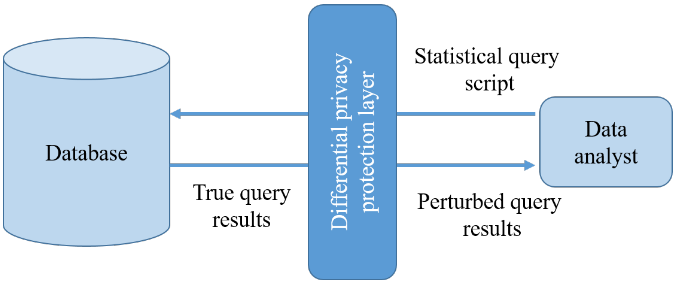

Fig. 1. Usage of differential privacy

3.1. Differential Privacy

Differential privacy prevents direct access to sensitive data in its raw form. Instead of direct access, the data analyst is provided with a statistical query interface. The mechanism is outlined in Fig. 1. According to the figure, (i) the analyst submits a statistical query script to the differential privacy (DP) protection layer, (ii) the DP layer computes the sensitivity of the submitted query script, (iii) the DP layer obtains true query results from the database, and (iv) the DP layer adds noise to the original results according to the query script's sensitivity computed in step (ii).

The magnitude of the noise added to true query results depend on the "sensitivity" of the query script. Sensitivity of a script is defined over pairs of databases that differ in only one record. Such databases are called sibling databases [6]. The sensitivity of a query script is the maximum L1 distance among any sibling databases for the given query script. Please see Def. 3.1 for a formal discussion. Computing the sensitivity of a query script is known to be NP-hard [33].

Definition 3.1 (Sensitivity): Let Q be a query script. The sensitivity of Q, denoted Δ_Q is

$$\Delta_Q = \underset{siblings\ D,D'}{\text{argmax}} ||S^D(Q) - S^{D'}(Q)||_1 \qquad (1)$$

In Eq. 1, $S^D(Q)$ (resp. $S^{D'}(Q)$) represents the true response of database D (resp. D') to Q. The vector difference of these responses in L1 norm over any two siblings is the sensitivity of Q.

Based on Def. 3.1, sensitivity of Q is independent of the database used in the analysis. This means, sensitivity is a property of the query script and not the queried data. Sensitivity measures how detailed a query script is. If a single-record change between two siblings can alter the true response heavily, then the sensitivity of Q will be computed higher.

Let us discuss a sample query script and compute its sensitivity: $Q = \{Q_1, Q_2\}$, where

- Q_1: `SELECT COUNT(*) FROM T WHERE Sex LIKE "Male"`
- Q_2: `SELECT COUNT(*) FROM T WHERE Age > 40`

Sibling databases D and D' will differ in only one record. Assume that these respectively are $D = \{r_1, r_2, \ldots, r_k\}$ and $D' = \{r_1, r_2, \ldots, r'_k\}$. As shown in Fig.2, depending on which region records r_k and r'_k reside, 16 different cases can be generated. Careful inspection of these is required:

- r_k and r'_k are in the same region: responses over D and D' would be the same.
- r_k is in region 1, r'_k is in region 2: responses to Q_2 would be the same. Q_1 responses differ by 1. Total distance is 1.
- r_k is in region 1, r'_k is in region 3: responses to Q_1 would be the same. Q_2 responses differ by 1. Total distance is 1.
- r_k is in region 1, r'_k is in region 4: responses to both Q_1 and Q_2 would differ by 1. Total distance is 2.

After analyzing all distinct cases, we conclude that the maximum possible L1 distance, therefore the sensitivity of the sample Q, is 2.

At this point, we are ready to define differential privacy concretely. Please refer to Def. 3.2 [6].

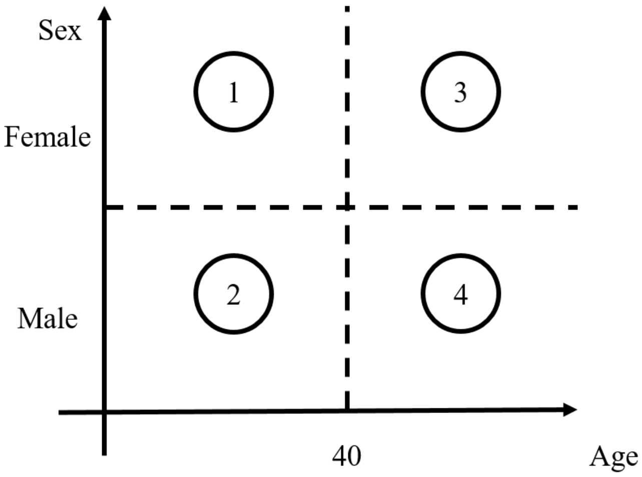

Fig. 2. Query regions of the sample query script

Definition 3.2 (Differential Privacy): Let D and D' be sibling databases and T be the space of all possible responses. Any randomized algorithm K satisfies differential privacy if

$$Pr[K(D) \in T] \leq e^{\epsilon} \times Pr[K(D' \in T] \qquad (2)$$

Many mechanisms that satisfy this definition has been proposed in the literature. In this work, we use the first and also the most popular mechanism by Dwork [6]. This mechanism, called the Laplace mechanism, perturbs true query responses with random noise picked from the Laplace distribution with 0 mean and at least Δ/ϵ magnitude. Higher levels of Δ and lower levels of ϵ imply heavier perturbation. Here, parameter ϵ represents the level of privacy protection offered by the mechanism.

3.2. Rating Prediction

The problem of utilizing the ratings matrix R to predict the possible rating a customer would give to an item if he/she had experienced the item is called “rating prediction”. A rating prediction method is evaluated by its prediction error. Let us suppose that the rating scale is the range [1-5]. If customer C

rates item I as 3, whereas the predicted rating for the pair (C, I) is 3.5; rating prediction is said to be successful with $|3.5 - 3.0| = 0.5$ error. This value is called the absolute error in predicting (C, I).

Given a large set of pairs of customers and items, the prediction error accumulates. The measure "mean absolute error" (MAE) normalizes the accumulated error by the number of predictions made according to Eq. 3.

$$MAE = \frac{\sum_{i=1}^{n} |r_a^i - r_p^i|}{n} \tag{3}$$

In Eq. 3, r_a^i denotes the actual rating given by the customer on the corresponding item and similarly, r_p^i denotes the prediction for the same. Here, n represents the number of predictions made during the evaluation.

There are two basic approaches to producing r_p^i given the prediction task (C, I): customer-based, or item-based. Customer-based approaches inspect the rating row of customer C, denoted $R[C]$, to find k other customers that are close to C in their own rating rows. These similar/close customers are called *neighbors* of C. Customer-based approaches then look at the ratings of the neighbors on item I and return a weighted average rating based on the similarity between C and his/her neighbor C'. The entire process involves accessing the rating matrix R in row-order and yields customer similarities. As a result, customer-based rating prediction is not suitable for a privacy preserving solution. In this study, we resort to item-based prediction.

In contrast to customer-based approaches, item-based approaches rely on item-similarity scores. For the pair (C, I), prediction first identifies neighbors of item I and then takes a weighted average over the ratings of C on the similar items. Notice that item similarities are far less privacy sensitive compared to customer similarities and the average is taken over $R[C]$, which resides with C him/herself.

The inputs to rating prediction are as follows: ratings matrix R, customer-item pair (C, I) to be predicted, a measure of similarity across items S and the number of neighbors to be involved, k. Using these inputs, a baseline item-based rating predictor would yield r_p with Eq. 4.

$$r_p(C, I) = \frac{\sum_k S(I, I_k) \times R[C, I_k]}{\sum_k S(I, I_k)} \tag{4}$$

In Eq. 4, I_k represents a neighbor of item I. Neighbors are items that are closest to I with respect to the similarity measure S. Weighting is done by multiplying the rating C has given to I_k (i.e., $R[C, I_k]$) by the similarity $S(I, I_k)$ between I and I_k. Then the weighted sum is normalized by the sum of similarities of the neighbors.

There exist various alternatives for computing the item-similarity scores. Among these, we focus on the Cosine similarity because it has a geometric interpretation and also it is easy to compute the sensitivity of the Cosine similarity. Using our introduced notation, Cosine similarity can be defined as given in Eq. 5.

$$Cos(I, I') = \frac{\sum_{i=1}^{m} R[C_i, I] \times R[C_i, I']}{|R^T[I]| \times |R^T[I']|} \tag{5}$$

In Eq. 5, R^T is the transpose of ratings matrix R. Consequently, $R^T[I]$ and $R^T[I']$ represent the ratings column of item I and I' respectively. If there are m customers in total in R, then Cosine simply computes the scalar product of the two columns, normalized by the length.

4. Proposed Solution

We start with the sensitivity analysis of the Cosine similarity measure expressed in Eq. 5. A record of our database corresponds to a single customer's data. Consequently, sibling databases differ in a single row of ratings and not a single rating cell - as falsely assumed in related work [8].

A single-row change across sibling data sets D and D' imply that for an arbitrary pair of items I and I', the Cosine similarity between these items differ by at most 1. This is the maximum allowable difference, as the range of values for Cosine similarity is [0, 1]. Our formal analysis is detailed next.

Let the sibling data sets be denoted with D and D', and the items whose Cosine is being computed be denoted with I and I'. We consider the case where both siblings contain a single row, as this scenario yields the maximum effect of a single-row change.

In data set D, we set the ratings of both I and I' to be 5 - the maximum rating. Over D, the similarity between the items will be calculated as $\frac{5\times 5}{5\times 5} = 1$.

In data set D', we change the rating of I' from 5 to 0. The numerator of Cosine in Eq. 5 becomes $5 \times 0 = 0$. Over D', the Cosine similarity between I and I' will be calculated as 0.

Since the similarity can be 1 on D and 0 on D', we have shown that $\Delta \geq 0$. Adding to this the fact that Cosine is in the range [0, 1], we conclude that the sensitivity of querying the Cosine similarity between two arbitrary items is 1.

The pseudo-code of the proposed solution is given Alg. 1 below. The rating prediction method expressed in Eq. 4 requires picking k neighbors of the item I. This task of querying for neighbors cannot be modelled as a numeric query and this part of the solution requires employing the exponential mechanism of differential privacy. We thwart this problem by perturbing item-item similarity scores according to the sensitivity analysis given above and then picking the best k items for item I to be predicted randomly based on perturbed similarity scores. This solution essentially simulates the exponential mechanism.

Algorithm 1 Pseudo-code of the proposed solution

```
Require: Ratings matrix $R$, differential privacy parameter $\epsilon$, number of neighbors $k$, customer $C$, item $I$
Ensure: Predicted rating of $C$ on $I$ according to differential privacy
for Each item $I$ do
    for Each item $J$ do
        $Cos_J \leftarrow Cosine(I, J)$ over $R$
        Perturb $Cos_J$ with $Lap(0, 1/\epsilon)$
Sort items w.r.t. $Cos_J$
Pick the best $k$ items
return weighted average according to Eq. 4
```

4.1. Privacy Analysis

The proposed solution protects the ratings matrix R very strongly according to differential privacy. However, employing this solution requires a customer to expose all previous ratings he/she has given.

This restriction might cause concerns about individual privacy. Future work will focus on exactly this possible concern. A possible remedy could be to collect all prediction requests of a group of customers at an independent third party. Such an aggregation would protect customers from possible privacy threats of the service provider by disassociating item-similarity queries and customer-identities. Another advantage of the same solution would be the buffering of item-similarity scores. The third party could use the collective budget of its member customers to allocate their query budget much more efficiently.

5. Experimental Results

As explained in Sec. 3.2, recommendation systems are evaluated by their prediction error. In our experimental evaluation of the proposed solution, we use mean absolute error (MAE) defined in Eq. 3.

Alg. 1 has two primary parameters. These are the privacy parameter ϵ and the number k of neighboring items. The effects of k and ϵ to MAE in rating prediction are presented in Sec. 5.1 and Sec. 5.2 respectively.

In our experiments, we used the Netflix data set. This data set, commonly known in the literature as the "Netflix Prize Data Set" [34], has been shared by the media-services provider Netflix in a contest. In the contest, the developer team that beats Netflix's rating predictor was offered 1 million U.S.D. We believe the popularity of the contest and its large prize attests to the importance of recommendation engines. The Netflix data set contains 100,480,507 ratings from over 480,000 customers on 17,770 different movies.

It is very difficult to process a data set of this scale in a relational database. For this reason, we modelled the ratings matrix R as a graph over the graph database Neo4J [35]. According to this model, every customer and every item in the data set is represented with a vertex and a weighted edge connecting a customer with an item represents the rating given by the customer on the item.

Our experimental setup was as follows. We used the Netflix data set as our rating matrix R. In each experiment scenario, we randomly selected 1,000 non-zero ratings from R. A non-zero rating r_a implies an actual rating assigned by a customer C to an item I. Each such rating r_a has been erased from R. Then, we applied rating prediction according to Alg. 1 to obtain the predicted rating r_p and measured the mean absolute error (MAE) according to Eq. 3.

In each experiment scenario, our proposed solution is compared against the case of "no protection", which produces predicted ratings without any privacy protection according to Sec. 3.2. This simply means there is no perturbation (i.e., $\epsilon = \infty$), and no added error due to privacy constraints.

Obviously, when there is no privacy protection, there is no source of randomness and rating prediction is a deterministic process. However, our proposed solution is randomized. As a result, we repeat all experiments 10 times and report the average MAE measurements.

5.1. Effects of the Number of Neighboring Items

Increasing the number of neighboring items affects MAE in a couple of different ways. Firstly, as more neighbors are involved, the chances of picking the right neighbors (essentially, that of not picking the wrong ones) increases. On the other hand, as more and more neighbors are involved, the predicted rating moves farther away from being personal and tends to be closer to the respective customer's average rating value.

The results are depicted in Fig. 3. The series named "no protection" represent the MAE over original data, whereas the series named "DP" represents our proposed solution.

For the no protection case, notice that MAE first decreases slightly but then starts to increase. The initial decrease is due to having more neighbors involved - more neighbors imply a more balanced weighted average. However, once we exceed the op-

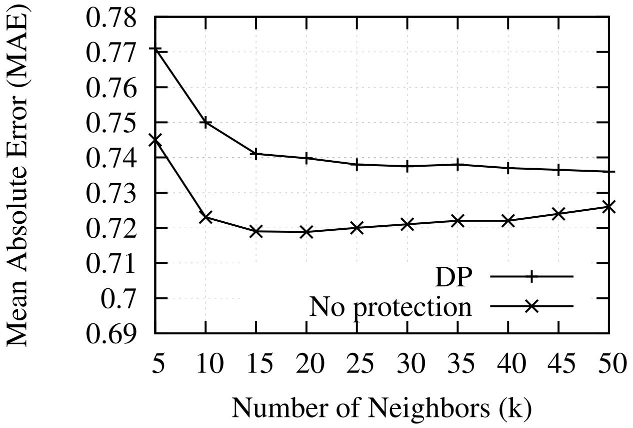

Fig. 3. The effects of varying number of neighbors k on MAE (ϵ = 0.5)

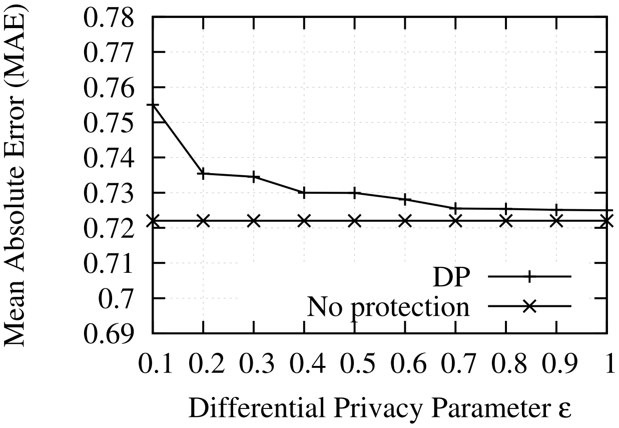

Fig. 4. The effects of varying ϵ on MAE (k = 35)

timal k, MAE increases because too many neighbors are involved and the prediction gets too close to the customer's average rating. The proposed solution is roughly 5% worse than the no protection case. Notice that this gap closes as k increases. This is because, prediction becomes insensitive to which neighbors are involved in the process.

5.2. Effects of the Differential Privacy Parameter

Recall that higher ϵ imply higher levels of noise. Quite naturally, we expect MAE to improve as ϵ increases. The results depicted in Fig. 4 verify this expectation.

According to the results, at high levels of protection such as $\epsilon = 0.1$, the difference in MAE is considerable. However, as ϵ increases, the gap closes and becomes almost unnoticeable. Obviously, MAE of the no protection case does not depend on ϵ.

6. Conclusion

In this study, we developed a basic, differentially private recommendation engine. Empirical results obtained over real-world data sets indicate that, every step taken towards protecting individual privacy has an adverse effect on prediction accuracy. However, such effects are limited: even with a very strong protection mechanism such as differential privacy, the rate of loss in prediction accuracy was measured to be below 5%, which we believe is a reasonable trade-off for privacy protection.

In future work, we plan to investigate the possible gains to be obtained with other differential privacy mechanisms such as the exponential mechanism.

Acknowledgments

This work is supported by the Scientific Research Project Fund of Adana Alparslan Turkes Science and Technology University under the project number 16103007.

References

[1] "Booking.com," http://www.booking.com, accessed: 2019-12-03.

[2] "The internet movie database," http://www.imdb.com, accessed: 2019-12-03.

[3] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender systems handbook*. Springer, 2011, pp. 1–35.

[4] Z. Batmaz and H. Polat, "Randomization-based privacy-preserving frameworks for collaborative filtering," *Procedia Computer Science*, vol. 96, pp. 33–42, 2016.

[5] A. J. Jeckmans, M. Beye, Z. Erkin, P. Hartel, R. L. Lagendijk, and Q. Tang, "Privacy in recommender systems," in *Social media retrieval*. Springer, 2013, pp. 263–281.

[6] C. Dwork, "Differential privacy: A survey of results," *Theory and Applications of Models of Computation*, vol. 4978, 2008.

[7] P. Hustinx, "Privacy by design: delivering the promises," *Identity in the Information Society*, vol. 3, no. 2, pp. 253–255, 2010.

[8] F. McSherry and I. Mironov, "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, 2009, pp. 627–636.

[9] J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov, "" you might also like:" privacy risks of collaborative filtering," in *2011 IEEE Symposium on Security and Privacy*. IEEE, 2011, pp. 231–246.

[10] I. Gunes, and H. Polat, "Robustness analysis of privacy-preserving hybrid recommendation algorithm," *International Journal of Information Security Science*, vol. 4, no. 1, pp. 13–25, 2015.

[11] I. Gunes and H. Polat, "Detecting shilling attacks in private environments," *Information Retrieval Journal*, vol. 19, no. 6, pp. 547–572, 2016.

[12] B. D. Okkalioglu, M. Koc, and H. Polat, "Reconstructing rated items from perturbed data," *Neurocomputing*, vol. 207, pp. 374–386, 2016.

[13] H. Polat and W. Du, "Achieving private recommendations using randomized response techniques," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2006, pp. 637–646.

[14] H. Polat and W. Du, "Privacy-preserving collaborative filtering on vertically partitioned data," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2005, pp. 651–658.

[15] H. Polat and W. Du, "Privacy-preserving top-n recommendation on horizontally partitioned data," in *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*. IEEE, 2005, pp. 725–731.

[16] M. Okkalioglu, M. Koc, and H. Polat, "A Privacy Review of Vertically Partitioned Data-based PPCF Schemes," *International Journal of Information Security Science*, vol. 5, no. 3, pp. 51–68, 2016.

[17] S. Zhang, J. Ford, and F. Makedon, "Deriving private information from randomly perturbed ratings," in *Proceedings of the 2006 SIAM international conference on data mining*. SIAM, 2006, pp. 59–69.

[18] A. Yargic, and A. Bilge, "Privacy-preserving multi-criteria collaborative filtering," *Information Processing and Management*, vol. 56, no. 3, pp. 994–1009, 2019.

[19] A. Boutet, D. Frey, R. Guerraoui, A. Jégou, and A.-M. Kermarrec, "Privacy-preserving distributed collaborative filtering," *Computing*, vol. 98, no. 8, pp. 827–846, 2016.

[20] J. Li, J.-J. Yang, Y. Zhao, B. Liu, M. Zhou, J. Bi, and Q. Wang, "Enforcing differential privacy for shared collaborative filtering," *IEEE Access*, vol. 5, pp. 35–49, 2016.

[21] Z. Erkin, M. Beye, T. Veugen, and R. L. Lagendijk, *Privacy-preserving content-based recommender system*. ACM, 2012.

[22] J. Canny, "Collaborative filtering with privacy," in *Proceedings 2002 IEEE Symposium on Security and Privacy*. IEEE, 2002, pp. 45–57.

[23] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh, "Privacy-preserving matrix factorization," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 801–812.

[24] Y. Shen and H. Jin, "Privacy-preserving personalized recommendation: An instance-based approach via differential privacy," in *2014 IEEE International Conference on Data Mining*. IEEE, 2014, pp. 540–549.

[25] Y. Shen and H. Jin, "Epicrec: Towards practical differentially private framework for personalized recommendation," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. ACM, 2016, pp. 180–191.

[26] H. Shin, S. Kim, J. Shin, and X. Xiao, "Privacy enhanced matrix factorization for recommendation with local differential privacy," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1770–1782, 2018.

[27] H. Zhou, G. Yang, Y. Xu, and W. Wang, "Effective matrix factorization for recommendation with local differential privacy," in *International Conference on Science of Cyber Security*. Springer, 2019, pp. 235–249.

[28] S. Zhang, L. Liu, Z. Chen, and H. Zhong, "Probabilistic matrix factorization with personalized differential privacy," *Knowledge-Based Systems*, vol. 183, no. 104864, pp. 1–11, 2019.

[29] X. Meng, S. Wang, K. Shu, J. Li, B. Chen, H. Liu, and Y Zhang, "Towards privacy preserving social recommendation under personalized privacy settings," *World Wide Web*, vol. 22, no. 6, pp. 2853–2881, 2019.

[30] R. Guerraoui, A.-M. Kermarrec, R. Patra, and M. Taziki, "D 2 p: distance-based differential privacy in recommenders," *Proceedings of the VLDB Endowment*, vol. 8, no. 8, pp. 862–873, 2015.

[31] E. O. Turgay, T. B. Pedersen, Y. Saygın, E. Savaş, and A. Levi, "Disclosure risks of distance preserving data transformations," in *International Conference on Scientific and Statistical Database Management*. Springer, 2008, pp. 79–94.

[32] J. Ren, X. Xu, Z. Yao, and H. Yu, "Recommender systems based on autoencoder and differential privacy," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, 2019, pp. 358–363.

[33] X. Xiao and Y. Tao, "Output perturbation with query relaxation," *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 857–869, 2008.

[34] “Netflix prize data by kaggle,” https://www.kaggle.com/netflix-inc/netflix-prize-data/, accessed: 2019-12-03.

[35] “Neo4j graph platform,” http://www.neo4j.com/, accessed: 2019-12-03.