Privacy-preserving collaborative recommendations based on random perturbations

Nikolaos Polatidis*
Department of Applied Informatics, University of Macedonia, 54006, Thessaloniki, Greece

Christos K. Georgiadis
Department of Applied Informatics, University of Macedonia, 54006, Thessaloniki, Greece

Elias Pimenidis
Department of Computer Science and Creative Technologies, University of The West of England, BS16 1QY, Bristol, United Kingdom

Haralambos Mouratidis
School of Computing, Engineering and Mathematics, University of Brighton, BN2 4GJ, Brighton, United Kingdom

Abstract

Collaborative recommender systems offer a solution to the information overload problem found in online environments such as e-commerce. The use of collaborative filtering, the most widely used recommendation method, gives rise to potential privacy issues. In addition, the user ratings utilized in collaborative filtering systems to recommend products or services must be protected. The purpose of this research is to provide a solution to the privacy concerns of collaborative filtering users, while maintaining high accuracy of recommendations. This paper proposes a multi-level privacy-preserving method for collaborative filtering systems by perturbing each rating before it is submitted to the server. The perturbation method is based on multiple levels and different ranges of random values for each level. Before the submission of each rating, the privacy level and the perturbation range are selected randomly from a fixed range of privacy levels. The proposed privacy method has been experimentally evaluated with the results showing that with a small decrease of utility, user privacy can be protected, while the proposed approach offers practical and effective results.

Keywords Collaborative filtering, Random perturbations, Multi-level privacy, Recommender systems

* Corresponding author (Email: npolatidis@uom.edu.gr, Tel: +302310891810)

Email Addresses (Nikolaos Polatidis: npolatidis@uom.edu.gr, Christos K. Georgiadis: geor@uom.edu.gr, Elias Pimenidis: Elias.Pimenidis@uwe.ac.uk, Haralambos Mouratidis: H.Mouratidis@brighton.ac.uk)

1. Introduction

Recommender systems aim to solve the information overload problem found in various online environments such as e-commerce and social networks (Polatidis and Georgiadis 2016; Shi et al., 2014; Moradi and Ahmadian 2015). Such systems can be used to recommend products, services, or users to users both implicitly or explicitly. The most used recommendation method is collaborative filtering (Shi et al., 2014; Ekstrand et al., 2011; Ar and Bostanci 2016).

Collaborative filtering is a recommendation method where a database of ratings is utilized and a similarity measure is used to make predictions based on ratings provided by other registered users (Shi et al., 2014; Konstan and Riedl, 2012). A database of ratings is essential and an example of such is shown in table 1.

	Product 1	Product 2	Product 3
User 1	3	2	-
User 2	-	5	4
User 3	3	4	4
User 4	4	3	3

Table 1. A ratings database

Considering that a user wants recommendations, a similarity function such as the Pearson Correlation Similarity, is used to create a neighborhood of the most similar users for the user who is requesting the recommendations. Using the Pearson function the statistical correlation between two users is calculated and a value between -1 to 1 is returned. Pearson correlation is shown in equation 1 and is the most frequently method used in collaborative filtering (Ekstrand et al., 2011). P is the set of all products, Sim (a, b) is the similarity between two users a and b, $r_{a,p}$ is the rating of user a for product p, $r_{b,p}$ is the rating of user b for product p, and $\bar{r}a, \bar{r}b$ represent the users' average ratings.

$$Sim\ (a,b) = \frac{\sum p \in P(r\ (a,p) - (\bar{r}a))(r\ (b,p) - (\bar{r}b))}{\sqrt{\sum p \in P(\ r\ (a,p) - (\bar{r}a))^2}\sqrt{\sum p \in P(\ r\ (b,p) - \ (\bar{r}b))^2}} \quad (1)$$

The recommendations are generated based on rating predictions of how likely a user is to prefer items that they have not looked at yet. These are based on historical ratings common with those of other users. The algorithm will compute the degree of similarity using a function such as that of Pearson between the user who is requesting the recommendations and those of other users. A neighborhood of the most similar users is created consisting of those with the higher degree of similarity. As a last step, rating predictions for unobserved items are generated between previous preferences of the user and those of the neighbors and the items with the highest rating predictions are recommended.

Privacy is an important issue for users of such systems (Bilge et al., 2013; Ozturk and Polat, 2015). In the context of ratings, privacy concerns make users unwilling to submit ratings, thus leading to sparsely populated relevant datasets which in turn can lead to lower degrees of similarity and eventually to poor recommendations. In typical scenarios, a recommendation system which is based on the client-server model, accepts requests from users and responds with recommendations. The ratings are submitted

directly from the client to the server. Users need to be registered in order to receive personalized recommendations. As submitted ratings are directly linked to individual users, the privacy of the ratings is considered an important aspect (Berkovsky et al., 2012; Jeckmans et al., 2013; Toch et al., 2012; Kobsa, 2007). In the context of our work, privacy is particularly related to the protection of the ratings. To avoid data leakage, and given the fact that the ratings are one of the most important information found in collaborative filtering these should be protected by using an appropriate privacy-preserving system (Kobsa, 2007). The perturbation of ratings is often utilized to achieve the desired level of privacy. The two most common cases where perturbation of ratings is essential in ensuring privacy are: a) data release, where a subset of stored ratings is transferred to a user's personal computer / device for local processing and b) where employees could be in a position of exploiting their access rights to registered users' private information.

Our approach uses a multi-level method aiming in protecting the privacy of the ratings. The following contributions have been made:

- A privacy-preserving multi-level method that perturbs the ratings of the users before they are submitted to the server is introduced.
- The proposed method is experimentally evaluated using five real datasets. It is shown that our method is both practical and effective.

The rest of the paper is organized as follows: Section 2 presents related work, Section 3 describes the proposed method, Section 4 explains the experimental evaluation and Section 5 contains the conclusions and proposals for future work.

2. Related work

In collaborative filtering, there are privacy concerns about user ratings that can be collected by the service provider or untrusted third-parties. Due to such concerns, users may not be willing to submit ratings or might submit fake ratings, thus, resulting in recommendations with poor relevance. Thus, generating as accurate recommendations as possible, while preserving user privacy is a serious challenge. According to Shyong et al. (2006), there are three main threats that may cause a collaborative filtering method not to work as expected and are the following:

1. The undesired access to private data by untrusted parties.
2. The manipulation of private user profiles in order to recommend certain products or services.
3. The service denial or malfunctioning of the system.

The two main approaches that can be utilized for privacy-preservation of personal user data such as the ratings are:

Centralized: Where all data are stored in a single server.
Decentralized: Where the data are distributed in more than one location and/or server.

In the case of centralized approach, there are a number of different methods for privacy-preserving recommendations:

A classical approach for privacy-preserving collaborative filtering is that of rating modification. Polat and Du (2005) developed a randomized perturbation technique, which perturbs every rating before it is submitted to the server. In their method, the perturbation value is derived from a distribution. Another privacy-preserving collaborative filtering approach that is based on a bisecting k-means algorithm is proposed by Bilge and Polat (2013). In this approach the authors propose a preprocessing scheme that is

based on two stages. Initially the algorithm uses a binary decision tree while in the second step it creates clones of users by injecting pseudo predictions in the original user data. Kikuchi and Mochizuki (2012), proposed a method for privacy-preserving collaborative filtering that adds random noise to the original rating data and then uses a posterior probability distribution method based on Bayes for reconstructing the original distribution of ratings. An interesting approach that uses data obfuscation to provide privacy-preserving collaborative filtering is found in Parameswaran and Blough (2007). In this method, recommendations are generated by combining data from multiple sources and obfuscating them before sending them to a centralized database. Additional works based on data modification include the one offered by Zhang et al. (2006). In this work, an agreement is established between the server and the users regarding the disclosure measure and the server sends guidelines to the user for modifying the data before submission. A different approach offering to protect ratings in a form of k-anonymity can be found in Casino et al. (2015). A number of *k* clusters of users is created, with each cluster having the same ratings and value for each of the clusters. This method is used when all the ratings are available in a centralized database and need to be released. Zhu et al. (2014) use differential noise to protect user privacy by providing nearest neighborhood attack resistance. The noise is added at the produced similarity value in order to avoid attacks from people who observe the generated recommendations and thus, can guess what ratings to submit to affect the generated output. An interesting approach has been proposed by (Zhang et al., 2014). In this method, the authors use a combination of a uniform and Gaussian distribution to perturb a rating before it is submitted to the server. After the submission of the perturbed rating an intensity weight is also submitted to the server. This intensity weight assists the server to produce results of higher accuracy when compared to simple perturbation methods.

An alternative direction in providing privacy-preserving recommendations is through the means of distributed storage, where attackers need to gain access to multiple databases instead of centralized one and the guidelines for generating recommendations using forms of distributed computing are outlined in Tveit (2001). In "PocketLens", Miller et al., (2004) show that the performance of distributed collaborative filtering is close to that of centralized systems. Canny (2002) proposed a method where the users have control of their data and are grouped into communities. In this method, when recommendations were requested, all the data of people who form the community was combined into one output and individuals are more protected. Aimeur et al. (2008) use a cryptographic approach when the client is communicating with the server and the use of a semi-trusted third party is proposed. Shokri et al. (2009) proposed to enlarge a user profile with other similar profiles, using a distributed mechanism, before sending any data to the server. In this approach, every user stores her profile offline and merges it partly with profiles of similar users after direct contact with them. An alternative decentralized approach can be found in Kaleli and Polat (2010), where a community of people is used to create a peer to peer network.

In addition to these two categories of approaches, other relevant privacy-preserving methods exist and include techniques that have been listed in Kobsa (2007) and Toch et al. (2012). These make use of pseudonymous and anonymous user modelling, client-side personalization and encrypted aggregation. A rather interesting approach is that of personalization of privacy, where each user has a personalized plan regarding his privacy level, which results to different perturbation levels (Kobsa, 2007).

Although, the related works are interesting and add their unique values to the literature, our proposed method described in section 3 adds its own characteristics by introducing randomization levels. Multiple levels in privacy-preserving collaborative filtering can be used by other methods too. Multi-level privacy brings an extra level of confusion to potential attackers since it makes it harder to guess the real rating value. Furthermore, attackers or people who can gain access to the database of ratings in order to affect the output by executing a KNN attack will find this more difficult, since (a) the rating values are perturbed and (b) future submitted ratings will be perturbed after a two-step randomization algorithm takes place at the client side.

3. Proposed method

In this paper, we propose a centralized approach, focused on rating privacy. Our method is based on the personalization of privacy, and from that point of view our approach is influenced by the work of Kobsa (2007), as we are also offering a solution that uses different perturbation levels. In our method privacy level and, thus, the perturbation range is created randomly for each submitted rating. Therefore, each perturbed rating discloses no information about the actual perturbation range that has been used, thus making it harder to guess the real value.

We propose the use of a multi-level method for privacy-preserving collaborative filtering. In our privacy-preserving recommendation system we utilize the insertion of a random value to the actual rating before it is submitted to the server. Randomization is a widely-used privacy preservation method in privacy-preserving data mining (Aggarwal et al., 2008). Furthermore, the benefit of using a simple logic algorithm for perturbation is that it can be easily installed in the client side and that is less complex to use and understand. A high-level overview of our approach is shown in figure 1.

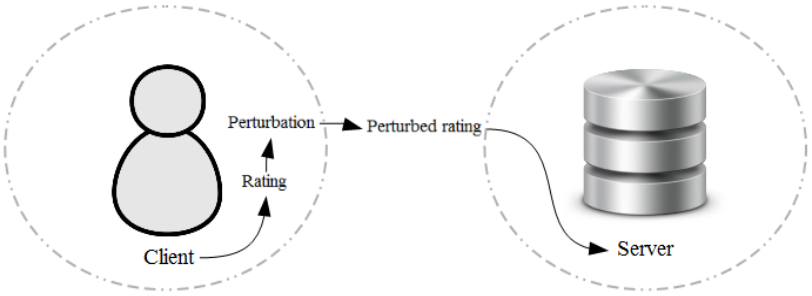


Figure 1. Privacy-preserving rating submission

In our method, the insertion of multiple privacy levels (with each level perturbing the rating with a different range of values), adequately protects user privacy while maintaining an acceptable level of accuracy. Algorithm 1 describes the proposed perturbation method. The values of MINRATING and MAXRATING refer to the rating scale used by the recommender system. For example, in the case that we have a scale in the range 1 to 5 then MINRATING refers to 1 and MAXRATING refers to 5. If a value, due to the perturbation method, drops below MINRATING then the perturbed rating takes the value of MINRATING and in the case that it exceeds MAXRATING then the perturbed rating takes the value of MAXRATING.

Algorithm 1: Privacy (Runs on the client)

Input: Rating
Output: PerturbedRating

```
Integer Level = Generate Random [1…n] // Privacy levels from 1=low, to 2=medium to 3=high, to n=higher
Integer Rand // Random integer [-t…t]
If (Level=1)
Generate random value Rand from [-1…1] // -1, 0 or 1
Perturbed Rating = Rating + Rand
        If (PerturbedRating < MINRATING)
                PerturbedRating = MINRATING
```

```
        Else If (PerturbedRating >MAXRATING)
                PerturbedRating = MAXRATING
        End If
End If
Else If (Level=2)
Generate random value Rand from [-2…2] // -2, -1, 0, 1 or 2
Perturbed Rating = Rating + Rand
        If (PerturbedRating < MINRATING)
                PerturbedRating = MINRATING
        Else If (PerturbedRating >MAXRATING)
                PerturbedRating = MAXRATING
        End If
End Else If
Else If (Level=3)
Generate random value Rand from [-3…3] // -3, -2, -1, 0, 1, 2 or 3
Perturbed Rating = Rating + Rand
        If (PerturbedRating < MINRATING)
                PerturbedRating = MINRATING
        Else If (PerturbedRating >MAXRATING)
                PerturbedRating = MAXRATING
        End If
End Else If
…
Else If (Level=n)
Generate random value Rand from [-n…n] // -n, … -3, -2, -1, 0, 1, 2, 3, … n
Perturbed Rating = Rating + Rand
        If (PerturbedRating < MINRATING)
                PerturbedRating = MINRATING
        Else If (PerturbedRating >MAXRATING)
                PerturbedRating = MAXRATING
        End If
End Else If
Return PerturbedRating
```

4. Experimental evaluation

In this section, we experimentally evaluate our approach using five real datasets and widely used metrics with different parameters. The experiments were conducted on an Intel i3 2.13 GHz with 4GBs of RAM, running Linux. All the algorithms have been implemented using the Java programming language and the Recommender101[1] library (Jannach et al., 2013).

4.1 Real datasets

For the evaluation of our privacy-preserving method, we used different experimentation settings and five real datasets which are publicly available and widely used in evaluating recommenders. The datasets are

[1] http://ls13-www.cs.tu-dortmund.de/homepage/recommender101/index.shtml

MovieLens (Herlocker et al., 1999), MovieTweetings (Dooms et al., 2013), YahooMovies, YahooAudio and FilmTrust (Guo et al., 2013). Moreover, table 2 presents the basic statistics of the datasets.

- **MovieLens**

MovieLens is an online movie recommender system. The dataset contains 100 thousand ratings, with values from 1 to 5, of 1,682 movies from 943 users. The data have been collected from the University of Minnesota from their online movie recommender system.

- **MovieTweetings**

MovieTweetings is a dataset that has been crawled from Twitter users and is publicly available. The rating scale of the dataset is in the range 0 to 10. The dataset contains 431,780 ratings from 39,363 users for 22,610 items.

- **YahooMovies**

YahooMovies is a dataset obtained from Yahoo!, which is a part of the Yahoo! Webscope program. The rating scale of the dataset is in the range 1 to 13. The dataset contains 211,231 ratings, 11,915 movies and 7,642 users.

- **YahooAudio**

YahooAudio is a dataset obtained from Yahoo!, which is a part of the Yahoo! Webscope program. The rating scale of the dataset is in the range 1 to 5. The dataset contains 311,704 ratings, 1,000 songs and 15,400 users.

- **FilmTrust**

FilmTrust is a dataset crawled from the FilmTrust website. The rating scale of the dataset is in the range 0,5 to 4. The dataset contains 35,497 ratings, 2,071 movies and 1,508 users.

Dataset	Users	Items	Ratings	Rating scale
MovieLens	943	1,682	100,000	1 - 5
MovieTweetings	39,363	22,610	431,780	0 - 10
YahooMovies	7,642	11,915	211,231	1 - 13
YahooAudio	15,400	1,000	311,704	1 - 5
FilmTrust	1,508	2,071	35,497	0,5 - 4

Table 2. Dataset statistics

4.2 Accuracy measures

For measuring the accuracy of the generated recommendations of the proposed method we have used the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE). These metrics are widely accepted for evaluating recommender systems (Herlocker et al., 2004; Shani and Gunawardana, 2011; Jannach et al., 2010). MAE is defined in equation 2 with *pi* being the predicted rating and *ri* being the actual rating in the summation. MAE is used for computing the deviation between the predicted and the real ratings. Note that lower values mean better recommendation predictions. RMSE is shown in equation

3. RMSE is an equation that is similar to MAE but with squared values. In RMSE, lower values are better.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|pi - ri| \quad (2)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(pi - ri)} \quad (3)$$

4.3 Settings

For the experiments, the following settings have been used:

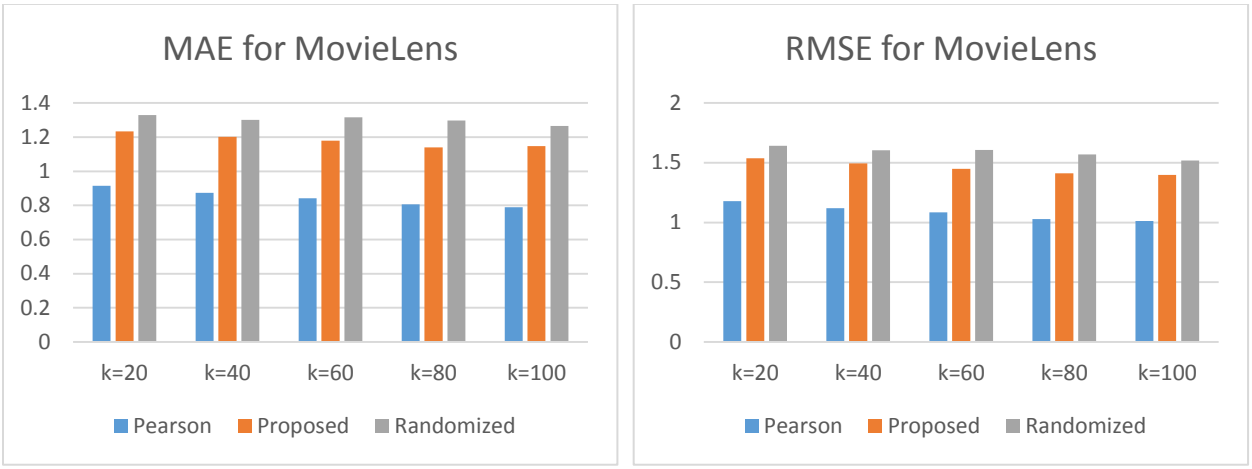
- **MovieLens dataset.** For this dataset, due to the rating scale being from 1 to 5 only the first two levels of the proposed method have been used.
- **MovieTweetings dataset.** For this dataset, the proposed method has used three levels.
- **YahooMovies.** For this dataset, four levels of the proposed method have been used.
- **YahooAudio.** For this dataset, the proposed method has used two levels.
- **FilmTrust.** For this dataset, the first two levels have been used.
- **MAE and RMSE.** For MAE and RMSE a 10-fold cross validation method has been used across all tests.
- **Pearson.** This is produced using collaborative filtering, equation 1 and the unaltered datasets.
- **Proposed method.** This is produced using collaborative filtering, equation 1 and the modified datasets having used the proposed method.
- **Randomized perturbations:** In this method, every rating is perturbed from a fixed range of values. For the MovieLens dataset this is from -2 to 2, for the MovieTweetings dataset this has been set from -3 to 3, for the YahooMovies dataset this is from -4 to 4, for the YahooAudio this is from -2 to 2 and for the FilmTrust dataset from -2 to 2. The fixed range of values for this algorithm have been selected because they are the same values used in the levels of the proposed method. This method is discussed in (Berkovsky, Kuflik, & Ricci, 2012).

However, it should be noted that the range of random value insertion has been chosen after experimentation with different values. The selected fixed values provided the best results between accuracy and privacy protection for each of the datasets. Although, different fixed ranges or random ranges of numbers can be used in all methods if the perturbed rating is within the scale used by the recommender system. In any case experiments need to take place in order to verify that accurate recommendations can still be provided.

4.4 Results

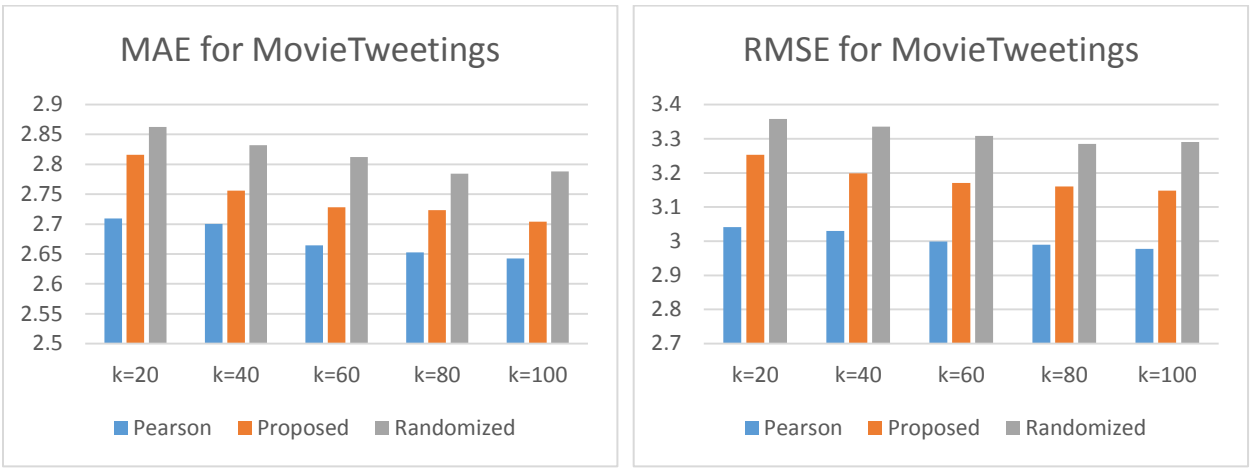
The MAE and RMSE results obtained from the MovieLens dataset are shown in figure 2. In figure 3 the results from the MovieTweetings dataset are shown. In figures 4 and 5 the results from the YahooMovies

and YahooAudio results are shown and in figure 6 the results from the FilmTrust dataset are shown. In all figures, sub-figure (a) represents the MAE results and sub-figure (b) represents the RMSE results.



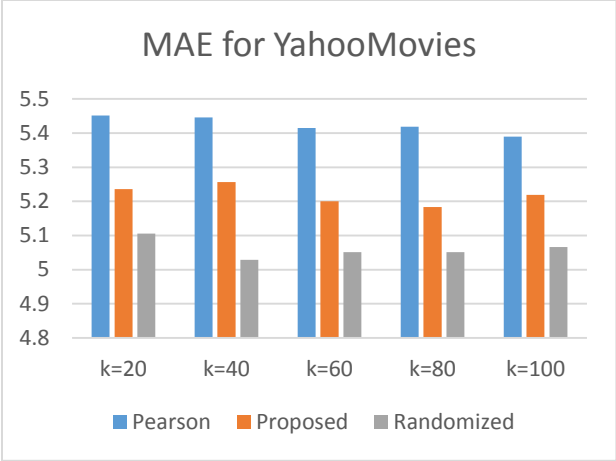
(a) (b)

Figure 2. Accuracy results for MovieLens

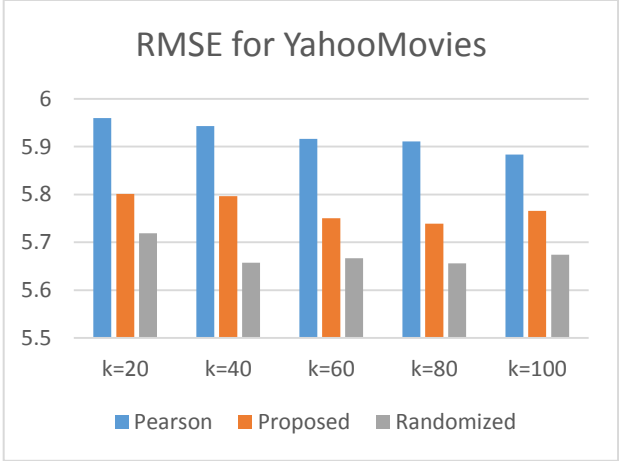


(a) (b)

Figure 3. Accuracy results for MovieTweetings

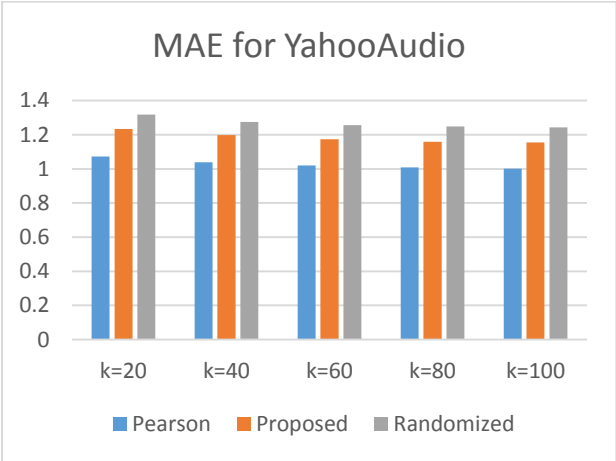


(a)

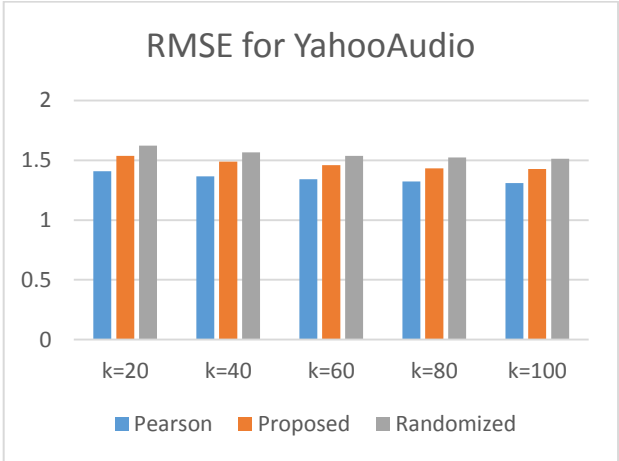


(b)

Figure 4. Accuracy results for YahooMovies



(a)



(b)

Figure 5 Accuracy results for YahooAudio

(a) (b)

Figure 6. Accuracy results for FilmTrust

4.5 Protection assessment

In evaluating the proposed method for privacy protection, we have used the sum of squared errors (SSE), which is a statistical method that measures information loss. SSE is used as a measure of distortion on the original data and has also been used as a measure in privacy-reserving collaborative filtering (Domingo-Ferrer and Mateo-Sanz, 2002; Casino et al., 2015). SSE is shown in equation 4. O is the original unaltered rating n *x* m matrix derived from the unaltered dataset with O_{ij} being its elements and P is the n *x* m matrix of the perturbed dataset with P_{ij} being its elements. Furthermore, a zero value is returned, which means that there is no distortion and therefore no protection.

$$SSE = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(O_{ij} - P_{ij}\right)^2 \quad (4)$$

Table 3 shows the results obtained for the datasets. For simplicity reasons the values have been converted to the 10^4 scale. It shown that the differences are mostly similar which means that in all cases the values are far from zero and the user privacy is protected.

Perturbation method	**Datasets**				
	MovieLens	**MovieTweetings**	**YahooMovies**	**YahooAudio**	**FilmTrust**
Proposed Method	14	32	43	23	13
Randomized perturbations	17	33	50	15	12

Table 3. SSE results

Furthermore, alternative methods such as the value difference (VD) described in Xu et al., (2005) can be used to measure information loss. Table 4 shows the results obtained using the VD method.

Perturbation method	Datasets				
	MovieLens	**MovieTweetings**	**YahooMovies**	**YahooAudio**	**FilmTrust**
Proposed method	0.56	0.30	2.6	0.08	0.50
Randomized perturbations	0.33	0.81	2.0	0.06	0.47

Table 4. VD results

When comparing VD with SSE it is shown that both assist in the protection assessment of different perturbation methods by showing that there are differences in the rating values between the methods. Furthermore, the numerical range of the results obtained using the VD method are making it easier to understand the protection level. Despite the fact that these methods asses in different ways the alterations of the datasets, MAE and RMSE metrics, are widely used to measure accuracy in terms of rating prediction (as it has been analyzed in previous paragraphs).

4.6 Discussion

Our proposed privacy-preserving recommendation algorithm is an important step for developing recommender systems adopted by users with privacy concerns. We consider the interesting part of the proposed method the use of random perturbations within different multiple levels. For example, when using plain randomized perturbations, also discussed in (Berkovsky et al., 2012), a random value is added to the rating and is usually derived from a distribution. On the other hand, we extend this approach by introducing multiple-level privacy protection based on random perturbations. Initially a privacy level is created and then the random value that will be added or subtracted from the rating is generated randomly with a value from within the level. However, other methods such as (Jingqi Zhang, Jianming Zhu, 2014), introduce the concept of privacy-preserving intensity weight. This is a value that is sent to the server with the perturbed rating in order to enchase the similarity value when the server produces the recommendations. Furthermore, in centralized architectures there are concerns about data release and the method proposed by Casino et al., (2015) fills this gap by proposing a method based on k-anonymity for releasing rating data. This method assumes that the submitted ratings are unaltered and is used only for privacy-preserving data release. Table 5 provides a comparison between our proposed method and state-of-the-art alternative perturbation methods.

Perturbation method	**Multiple Levels**	**Different protection range**	**Maintain high accuracy**
Proposed method	Yes	Yes	No
Randomized	No	Yes	No
Zhang et al., (2014)	No	Yes	Yes
Casino et al., (2015)	No	Yes	No

Table 5. Perturbation method comparison

Different perturbation methods can be used according to the scope of each method for privacy-preserving recommendations. Moreover, each method needs to be evaluated according to the scenario that it will be

applied. We run several experiments and the results show that, for every dataset, except the YahooMovies dataset, after the proposed perturbation method takes place the accuracy level is still usable and as the neighborhood grows the difference between the methods is similar. Furthermore, when using alternative experimentation settings is shown that the difference in accuracy between the unaltered datasets and the perturbed datasets remain very much alike as the previously used settings. Thus, we can conclude that with a small decrease in accuracy the privacy of the users is protected and still accurate recommendations can be provided. Although, it is remarkable that in the YahooMovies dataset we have the opposite result than the expected (the accuracy is improved). Furthermore, when comparing our proposed method with an alternative and with similar perturbation settings, it is shown that for every dataset when the ratings are perturbed with each of these methods the results are quite close. Although, if the perturbation range changes, then the output could be different, resulting in an unbalanced system that could either offer less protection with higher accuracy levels or high protection with lower accuracy levels. Therefore, a balance needs to be maintained and tests need to take place when deciding the range of the values that will be used from the perturbation method in order to have a usable system that protects privacy.

Besides the accuracy measurements when using privacy-preserving collaborative filtering systems, statistical methods such as SSE can be used to evaluate the protection offered. In our experiments, we have used SSE and VD metrics to evaluate the protection offered by our method and an alternative. In SSE and VD, if we compare two identical datasets then the result returned is zero. Therefore, no protection is provided if a zero value is the output. Consequently, in the results it is shown that the values of our method and the alternative are distant from zero and that privacy is protected. Thus, different evaluation methods need to be applied to have more concrete results when it comes to privacy-preserving collaborative filtering.

We have proposed a multi-level privacy preservation method for collaborative filtering recommender systems. Our primary intention is the introduction of multiple-levels in the perturbation process. We aim to introduce the concept of multiple levels to the practitioner. However, there are certain implications that need to be considered before developing a multi-level privacy preservation system and include:

1. How many levels to use.
2. If the range within each level will be fixed or random.
3. The accuracy is relevant to the levels, the perturbation range and the rating scale.
4. Several experiments need to take place to verify the necessary number of levels and perturbation range, in order to maintain a reasonable tradeoff between accuracy and privacy.

5. Conclusions and future work

Even though collaborative recommender systems have matured as a concept, have found numerous practical applications in the business world and also inspired a good volume of research in academia, there are still privacy concerns from users of such systems. Users want useful recommendations, but at the same time are anxious about submitting ratings due to privacy concerns, thus leading to poor recommendations. One of the most successful means of protecting user privacy in collaborative filtering systems is to perturb the rating before it is submitted to the server. In our proposed method, we have used a multi-level perturbation method that perturbs each rating at the client side before it is submitted to the server. Experimental results demonstrate that our proposed method provides acceptable outcomes both in terms of accuracy, SSE and VD values, when compared to a similar alternative. Furthermore, different privacy protection measures provide different accuracy results, SSE and VD values, thus offering different protection levels. This is due to the fact that each method can be configured accordingly and each has its own unique characteristics.

Our method can be used in various online environments that are based on user ratings, to preserve user privacy by perturbing each rating before submission and provide recommendations of acceptable accuracy. The main achievement of our proposed method is that based on a randomly selected perturbation level for each rating, can lead any potential attacker to confusion since it becomes more difficult to guess the range of the perturbation for a specific rating. Additionally, is shown that our approach can preserve privacy, while the accuracy of the generated recommendations is of an acceptable level.

The main implication when applying the proposed method is the potentially negative impact found on the accuracy when generating recommendations. When applying a perturbation method, the ratings are altered and can usually differ from the real ones, thus leading to inaccurate computations of user similarity, which can lead to different nearest neighbors and to inaccurate rating predictions and eventually generated recommendations. Different perturbation methods might give different results, thus, protecting privacy at a different level.

In our future work, we aim to investigate the employability of collaborative filtering methods in other systems to provide recommendations. For example, collaborative filtering can be used as a part of a system that is used to provide privacy-preserving location-based services in mobile recommender systems. Another example is the tourism domain: mobile recommender systems have been utilized in the tourism domain and many tourists use such applications. Advances in mobile recommender systems for tourism that utilize collaborative filtering methods have become an active field of research and this will be another research direction for the future, with the proposal of privacy-preserving systems for mobile tourism.

Furthermore, we aim to concentrate on the robustness of collaborative filtering systems against shilling attacks. Malicious users may insert fake profiles into web systems in order to manipulate the recommendation results. It is indeed possible to develop profiles for the promotion of certain items by providing always high ratings for them and low values for others. This makes necessary the development of relevant detection mechanisms.

References

Aggarwal, C. C., & Philip, S. Y. (2008), *A general survey of privacy-preserving data mining models and algorithms*, Springer US.

Aimeur, E., Brassard, G., Fernandez, J. M., & Onana, F. S. M. (2008), "Alambic: a privacy-preserving recommender system for electronic commerce" International Journal of Information Security, Vol. 7 No. 5, pp. 307-334.

Ar, Y., & Bostanci, E. (2016). A genetic algorithm solution to the collaborative filtering problem. Expert Systems with Applications, 61, 122-128.

Berkovsky, S., Kuflik, T., & Ricci, F. (2012), "The impact of data obfuscation on the accuracy of collaborative filtering", Expert Systems with Applications, Vol. 39 No. 5, pp. 5033-5042.

Bilge, A., & Polat, H. (2013), "A scalable privacy-preserving recommendation scheme via bisecting k-means clustering", Information Processing & Management, Vol. 49 No. 4, pp. 912-927.

Bilge, A., Kaleli, C., Yakut, I., Gunes, I., & Polat, H. (2013), "A survey of privacy-preserving collaborative filtering schemes", *International Journal of Software Engineering and Knowledge Engineering*, Vol. 23 No. 8, pp. 1085-1108.

Canny, J. (2002), "Collaborative filtering with privacy" In Security and Privacy, 2002. Proceedings. 2002 IEEE Symposium on, IEEE, pp. 45-57.

Casino, F., Domingo-Ferrer, J., Patsakis, C., Puig, D., & Solanas, A. (2015), "A k-anonymous approach to privacy preserving collaborative filtering", Journal of Computer and System Sciences, Vol. 81 No. 6, 1000-1011.

Domingo-Ferrer, J., & Mateo-Sanz, J. M. (2002), "Practical data-oriented microaggregation for statistical disclosure control", *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 14 No. 1, pp. 189-201.

Dooms, S., De Pessemier, T., & Martens, L. (2013), "Movietweetings: a movie rating dataset collected from twitter", In Workshop on Crowdsourcing and human computation for recommender systems, CrowdRec at RecSys, Vol. 2013, p. 43.

Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2011), "Collaborative filtering recommender systems", Foundations and Trends in Human-Computer Interaction, Vol. 4 No. 2, pp. 81-173.

Guo, G., Zhang, J., & Yorke-Smith, N. (2013, August). A Novel Bayesian Similarity Measure for Recommender Systems. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI, pp. 2619-2625.

Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999), "An algorithmic framework for performing collaborative filtering", In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp. 230-237.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004), "Evaluating collaborative filtering recommender systems", ACM Transactions on Information Systems (TOIS), Vol. 22 No. 1, pp. 5-53.

Jannach, D., Lerche, L., Gedikli, F., & Bonnin, G. (2013, June). What recommenders recommend–an analysis of accuracy, popularity, and sales diversity effects. In *International Conference on User Modeling, Adaptation, and Personalization* (pp. 25-37). Springer Berlin Heidelberg.

Jannach, D., Zanker, M., Felfernig, A. & Friedrich, G. (2010), *Recommender Systems an Introduction*, Cambridge University Press.

Jeckmans, A. J., Beye, M., Erkin, Z., Hartel, P., Lagendijk, R. L., & Tang, Q. (2013), "Privacy in recommender systems", In Social media retrieval, Springer London, pp. 263-281.

Kaleli, C., & Polat, H. (2010). P2P collaborative filtering with privacy. *Turkish Journal of Electrical Engineering & Computer Sciences*, *18*(1), 101-116.

Kikuchi, H., & Mochizuki, A. (2012), "Privacy-preserving collaborative filtering using randomized response", In Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on, IEEE, pp. 671-676.

Kobsa A. (2007), "Privacy-Enhanced Web Personalization", The Adaptive Web, LNCS 4321, pp. 628-670.

Konstan J, Riedl J. (2012), “Recommender Systems: from algorithms to user experience”, User Modeling and User-Adapted Interaction Vol. 22, pp. 101-123.

Miller, B. N., Konstan, J. A., & Riedl, J. (2004), “PocketLens: Toward a personal recommender system”, ACM Transactions on Information Systems (TOIS), Vol. 22 No. 3, pp. 437-476.

Moradi, P., & Ahmadian, S. (2015). A reliability-based recommendation method to improve trust-aware recommender systems. Expert Systems with Applications, 42(21), 7386-7398.

Ozturk, A., & Polat, H. (2015), “From existing trends to future trends in privacy-preserving collaborative filtering”, *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 5 No. 6, pp. 276-291.

Parameswaran, R., & Blough, D. M. (2007), “Privacy preserving collaborative filtering using data obfuscation”, In Granular Computing, 2007. GRC 2007. IEEE International Conference on, IEEE, pp. 380-380.

Polat, H., & Du, W. (2005), “Privacy-preserving collaborative filtering” International Journal of Electronic Commerce, Vol. 9 No. 4, pp. 9-35.

Polatidis, N., & Georgiadis, C. K. (2016). A multi-level collaborative filtering method that improves recommendations. Expert Systems with Applications, 48, 100-110.

Shani, G., & Gunawardana, A. (2011), “Evaluating recommendation systems”. In Recommender systems handbook, Springer US, pp. 257-297.

Shi, Y., Larson, M., & Hanjalic, A. (2014), “Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges”, ACM Computing Surveys (CSUR), Vol. 47 No. 1, 3.

Shokri, R., Pedarsani, P., Theodorakopoulos, G., & Hubaux, J. P. (2009), “Preserving privacy in collaborative filtering through distributed aggregation of offline profiles”, In Proceedings of the third ACM conference on Recommender systems, ACM, pp. 157-164.

Shyong, K., Frankowski, D., & Riedl, J. (2006), “Do you trust your recommendations? An exploration of security and privacy issues in recommender systems”. In Emerging Trends in Information and Communication Security, Springer Berlin Heidelberg, pp. 14-29.

Toch E., Wang Y., Cranor L.F. (2012), “Personalization and Privacy risks and remedies in personalization-based systems”, User Modeling and User-Adapted Interaction Vol. 22, pp. 203-220.

Tveit, A. (2001), “Peer-to-peer based recommendations for mobile commerce”, In Proceedings of the 1st international workshop on Mobile commerce, ACM, pp. 26-29.

Xu, S., Zhang, J., Han, D., & Wang, J. (2005, May). Data distortion for privacy protection in a terrorist analysis system. In International Conference on Intelligence and Security Informatics (pp. 459-464). Springer Berlin Heidelberg.

Yahoo! Webscope dataset ydata-ymovies-user-movie-ratings-content-v1_0. [http://research.yahoo.com/Academic_Relations]

Yahoo! Webscope dataset ydata-ymusic-user-artist-ratings-v1_0 [http://research.yahoo.com/Academic_Relations]

Zhang, S., Ford, J., & Makedon, F. (2006), "A privacy-preserving collaborative filtering scheme with two-way communication", In Proceedings of the 7th ACM conference on Electronic commerce, ACM, pp. 316-323.

Zhang, J., Zhu, J., & Zhang, N. (2014). An Improved Privacy-Preserving Collaborative Filtering Recommendation Algorithm. In PACIS.

Zhu, T., Li, G., Pan, L., Ren, Y., & Zhou, W. (2014), "Privacy preserving collaborative filtering for KNN attack resisting", Social Network Analysis and Mining, Vol. 4 No. 1, pp. 1-14.