Achieving Optimal Privacy in Trust-Aware Social Recommender Systems

Nima Dokoohaki[1], Cihan Kaleli[2], Huseyin Polat[2] and Mihhail Matskin[1,3]

[1] Department of Electronics, Computer and Software Systems,
Royal Institute of Technology (KTH), 16440, Kista, Stockholm, Sweden
[2]Department of Computer Engineering, Anadolu University, Eskisehir, 26470, Turkey
[3]Department of Information and Computer Science,
Norwegian University of Science and Technology (NTNU), Trondheim, Norway
nimad@kth.se, ckaleli@anadolu.edu.tr, polath@anadolu.edu.tr , misha@kth.se

Abstract. Collaborative filtering (CF) recommenders are subject to numerous shortcomings such as centralized processing, vulnerability to shilling attacks, and most important of all privacy. To overcome these obstacles, researchers proposed for utilization of interpersonal trust between users, to alleviate many of these crucial shortcomings. Till now, attention has been mainly paid to strong points about trust-aware recommenders such as alleviating profile sparsity or calculation cost efficiency, while least attention has been paid on investigating the notion of privacy surrounding the disclosure of individual ratings and most importantly protection of trust computation across social networks forming the backbone of these systems. To contribute to addressing problem of privacy in trust-aware recommenders, within this paper, first we introduce a framework for enabling privacy-preserving trust-aware recommendation generation. While trust mechanism aims at elevating recommender's accuracy, to preserve privacy, accuracy of the system needs to be decreased. Since within this context, privacy and accuracy are conflicting goals we show that a Pareto set can be found as an optimal setting for both privacy-preserving and trust-enabling mechanisms. We show that this Pareto set, when used as the configuration for measuring the accuracy of base collaborative filtering engine, yields an optimized tradeoff between conflicting goals of privacy and accuracy. We prove this concept along with applicability of our framework by experimenting with accuracy and privacy factors, and we show through experiment how such optimal set can be inferred.

Keywords: Privacy, Trust, Optimization, Data Disguising, Social networks, Collaborative filtering, Recommender systems.

1 INTRODUCTION

Adaptive Web and its myriads of techniques are paving the path towards fulfilling the promise of alleviating classic problem of information overload. Recommenders, one of the most widely adopted and well-anticipated of this stack of technologies, remain the sole leader of this essential advancement. Recommenders intend to provide people with suggestions of products they will appreciate, based upon their past preferences, history of purchase, or demographic information [1]. Most successful recommenders employ well-known collaborative filtering (CF) techniques [2]. CF automates the word-of-mouth process, when asking like-minded friends or family members for their individual opinions on different matters like new movie releases. This process involves finding users similar to the user receiving the recommendation and

suggesting her items rated high in the past by similar taste users. Since there are always numerous items and the ratings scored by users are sparse, often the step of finding similar users fails. To alleviate this shortcoming, the former step was replaced by utilizing a trust metric, which enables a trust-based heuristic to propagate and spot users whom are trustworthy with respect to active user (*a*) that we are gathering recommendations for [14]. Recommenders that take advantage of fusion of interpersonal trust with CF heuristics within and across their architectures are collectively referred to as *Trust-Aware Recommender Systems* [16].

Privacy remains a foundational problem in personalization research. In general CF systems usually fail to protect users' privacy. Users who remain concerned about their privacy might use false data in the process. Using false data decreases accuracy of CF systems [7]. Users who are concerned about their privacy may employ false data, because data collected for CF can be used for unsolicited marketing, government surveillance, profiling users, misused, and it can be transferred [6]. As a matter of fact, it is more likely that users will give more truthful data if privacy measures are provided. Massa and Avesani [16] study the architecture and design of trust-aware recommender systems and describe how trust-aware recommenders alleviate shortcomings of traditional systems. Trust-aware recommenders are modeled and designed in a decentralized fashion. However, current implementations are either centralized or are not tested in a decentralized fashion [16]. As a result, there is a growing concern about the vulnerability of these systems to shilling attacks [28]. At the same time, as most research invested in analyzing trust-aware recommenders, focuses on improving the recommendations, they fail to clearly address the privacy issues surrounding the architecture and components of these systems. As a result this research work invests on dealing with privacy issues surrounding the architecture and components of trust-aware recommender systems.

To address these shortcomings, during the pace of this manuscript, we extend the architectural landscape of traditional CF techniques and trust-aware recommenders to include building blocks required for realizing a privacy-preserving trust-aware recommender system. As an example of such architecture, we implement a framework for applying data perturbation techniques to user rating profiles. To do this, we introduce a private trust computation process. Then, accordingly, we propose methods for producing private recommendations based on trust-based CF recommender systems. We ground this framework at the top of a social trust recommender system [19], which utilizes *T-index* [31] [32] as its trust metric. We will show how the overall trust estimation can be augmented to accommodate the private trust estimation and prediction generation. We design this framework, having protection and preserving users' privacy in mind, while still providing accurate recommendations on masked data using trust-enabled CF schemes. We conceptualize this tradeoff between accuracy and privacy as a Pareto frontier notion. We will show that privacy and trust mechanisms, each with their respective configurations jointly form configurations of the overall framework. According to Pareto optimality perspective, at least a joint setting of both configurations exists which when utilized results in privacy of user data being maintained, while keeping accuracy decent at the same time. To evaluate this framework, we study the accuracy of the

recommendations under different masked distributions and compare the results of the computations with original data.

Our experiment results clearly show that the proposed scheme provides recommendations with decent accuracy while preserving users' privacy. The rest of the manuscript at hand is organized, as follows: First, a background into the main concepts shaping the foundation of this work is presented. The architecture of the system is presented in the third section, followed by a detailed description of the approach. Experimental evaluation is presented in the forth section, followed by a discussion of results. Finally, a conclusion and future work brings this work to its respective end.

2 Background

2.1 Trust-Aware Collaborative filtering

CF algorithms generally make recommendations based on similarity between the users' tastes. Similarity measure is not sufficient when user rating scores are sparse and insufficient. In the face of these shortcomings, traditional user similarities deem useless and recommenders need new ways to calculate user similarity. As a response to this problem, interpersonal trustworthiness was proposed to replace old similarity measures. Ziegler et al. [18] describe a relationship between how similarity between two users can be interpreted as how much they might trust each other. Golbeck [13, 14] shows the correlation between similarity and trust and how it can elevate movie recommendation accuracy. Taking into account this fact, trust can be considered as a measure for expressing the relationship between two users in recommendation systems. O'Donovan and Smyth [17] approach trust-aware recommenders by utilizing a two-mode profiling model that documents the past behavior of users. Massa and Avesani [15, 16] present architecture for a trust-aware recommender system in which trust can be propagated and aggregated for all of the users in a social network setting. Lathia et al. [18] model a variation of *kNN* (*K-Nearest Neighbor*) CF recommender, which allows users to learn who and how much to trust by evaluating the utility of the rating information they receive. One of the problems with frameworks presented above is that the functionality of previous recommenders is dependent on availability of explicit trust ratings in between users to infer other trust relations.

Zarghami et al. [19] introduce a decentralized trust-aware recommender system, which utilizes *T*-index [33], as a trust metric for filtering trust between users. Unlike previous approaches, a trust network between users can automatically be built from existing ratings between users. Framework increases the probability of finding trustworthy users across the network by creating a Distributed Hash Table (DHT) like list of trustees, TopTrusteeList (TTL) [19] that wraps around the items, which are tasted similarly to those of current user. Our work utilizes this recommender as the foundation of our framework.

2.2 Privacy-Preserving Collaborative Filtering

Privacy remains the most significant problem in the context of CF recommendation systems. Canny [4, 5] proposes privacy-preserving schemes for CF. In his schemes, users control their private data and they are capable of getting personalized referrals produced without disclosing their data. Canny proposes to use homomorphic encryption in his schemes to protect individuals' privacy. Polat and Du [6] employ perturbation techniques to offer predictions. In their scheme, users disguise their private data before sending it to central server that collects masked data instead of actual data. Kaleli and Polat [7] study how to produce predictions while preserving individuals' privacy while producing naïve Bayesian classifier (NBC)-based private recommendations. They employ randomized response techniques (RRT) to protect users' privacy. Parameswaran [10] presents a data obfuscation technique in which she designs and implements a privacy-preserving shared collaborative filtering framework using data obfuscation algorithm. Berkovsky et al. [11] investigate a decentralized approach, which does not require sending data to a centralized server. Collaborative filtering techniques can be employed in the context of peer-to-peer (P2P) and social networks. Kaleli and Polat [6] propose a solution to produce NBC-based private recommendations in a P2P network. Authors present a solution to produce private referrals in a social network context [9]. Proposed solution requires using data disguising techniques.

Within the context of our framework, we have adopted this approach to provide private recommendations in the context of a trust network of users, where actual user profiles are masked and trust computation process and recommendation procedure are changed accordingly to produce private recommendations.

2.3 Preserving Privacy in Trust-Aware Recommender Systems

Taking measures for preserving privacy during trust calculation and computation has been of great importance. Lack of privacy protection within the context of systems dealing with trust and reputation, can *ease attacks by malicious insiders*, as they might infest the existing trust establishments or alter the trust computation results.

As a result, great deal of research has been invested in analyzing schemes for combining privacy with trust establishments in different fields. In Multi-Agent Systems, preserving privacy during trust negotiations between software agents in any open system is a crucial task because sensitive data is exchanged between strangers without any prior knowledge of each other [20, 21]. In P2P systems, similar concern for privacy is raised about the possibility that malicious users can exploit the peers trust network to spread tampered-with resources [22]. In the context of recommender systems, Lam et al. [23] give an overview of privacy and security problems with recommenders. These problems are twofold: the personal information collected by recommenders raises the risk of unwanted exposure and malicious users can bias or sabotage the recommendations that are provided to other users [23]. While former points out to *privacy of recommenders*, the latter is collectively referred to as *Shilling*

attacks [24]. Attacks on recommenders remain a significant security hole in these systems [16, 22]. As popularity of trust-aware recommenders in academic and industrial community increases, problem with attacks on trust-enabled recommenders remains at large. Zhang [25] executes an average shilling attack on a trust-aware recommender system and demonstrates that trust-recommender exhibits more stability over a traditional *kNN*-based recommender.

Our framework is designed with idea of being capable of withholding shilling attacks in mind. As the main focus of this work is on implementation and design of a privacy-preserving trust recommender, we leave the analysis of framework stability under different attacks for the future work.

3 Recommendation Framework

In this section, we present the framework that we have composed for building a private trust-aware recommender system. To do so, first, a brief introduction into the architecture and design of our trust-aware recommender is presented in the first section, while in the second section; we describe how this architecture can be extended with components needed to build a private trust recommender. This is followed by description of the process of trust estimation and prediction generation of our resulting system. In the last section we present the definition of optimal privacy set and the process of how to infer this set, with respect to the context of this work.

3.1 Architecture of a Private Trust-Aware Recommender System

Massa and Avesani [16] present a generic architecture for a trust-aware recommender system. This architecture is presented in Fig. 2.

In this architecture, gray boxes present modules, while white boxes represent the matrices used as input and outputs of algorithms. Typical inputs of the architecture are*: rating matrix* (rating scores assigned to items by users) and *trust network* [29] (trust statements of users with respect to each other). While rating is the main input of traditional CF recommenders, trust can be inferred and in our case, automatically generated out of the rating matrix. In this architecture, they visualize the anatomy of a traditional CF recommender being composed of two main building blocks: a *similarity metric* and a *rating predictor*. Similarity metric, helps finding similar users (or neighbors), which is typically *Pearson Correlation Coefficient* (PCC) [28]. Rating predictor module predicts ratings based on a weighted sum of ratings given by similar users to the items [16]. Architecture of a trust-aware recommender is made up of a *trust metric*. The difference between two architectures is made in how neighbors are discovered and how their weights are identified. This can be done through similarity module or through trust metric module. The combined output, estimated trust network along with user similarity, can be used to generation ratings predictions.

To introduce the notion of privacy within this architecture, we need to justify what we mean by privacy first. As we construct a trust network of users for propagation and aggregation of trust values within our framework, we propose for adoption of notion of social privacy across the network of users.

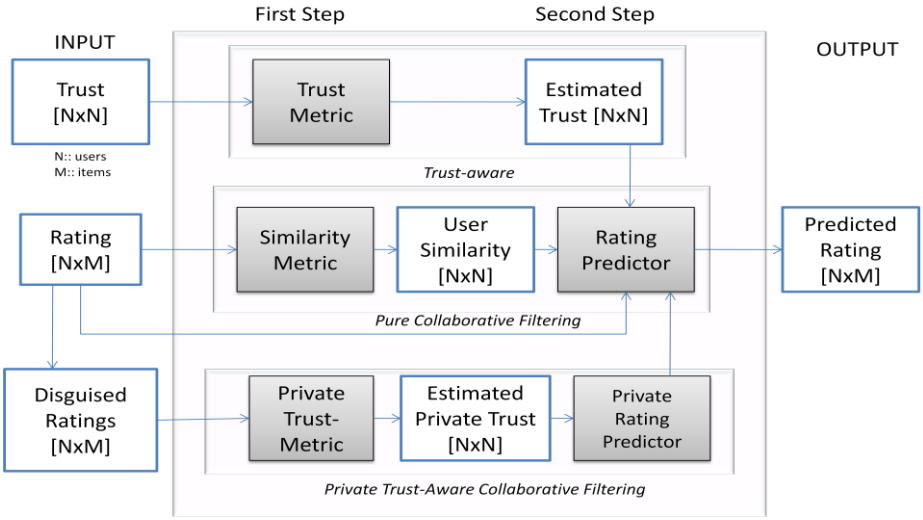


Fig. 1. *Architecture of a Private Trust-Aware Recommender System*. Trust-aware recommenders (top box) can be extended with privacy (bottom box) to enhance traditional similarity driven collaborative filterers (middle box). Computational modules are depicted in gray boxes, while inputs and outputs are depicted in white. Architecture is adapted from Massa and Avesani [16].

To define privacy in our terms, we approach the notion of privacy in following terms:

"Any user a, who wants prediction [in a Trust Network] does not have to reveal her rating vector during recommendation process and other users in the recommendation process cannot learn any rating value of a and the rated and/or unrated items of user a's rating vector".

Taking into account this adaptation of privacy, we have extended existing architecture in Fig.1 to include the architecture of a *private trust aware recommendation system*. As depicted, a privacy-preserving trust-aware CF can be composed of two main modules: a *private trust metric* and a *private rating predictor*. This architecture takes a masked rating set as input and generates a private trust estimation which is used by a private rating predictor, which in turn combined with rating prediction module from the pure CF step can generate predicted rating matrix. Obviously, a private trust recommender is actually composed of both pure and trust-enhanced recommendation modules and inputs. To understand how this architecture can be realized, we adapt this architecture onto our recommendation framework, presented in previous section.

In our framework, the private trust metric is realized through *data disguising* and *private trust calculation* steps.

To achieve our privacy aim, we propose to use *z-scores* of user ratings instead of their actual preferences. The z-score [6] of an item indicates how far and in what direction, that item deviates from its distribution's mean, expressed in units of its distribution's standard deviation. In this work we utilize z-score transformation for normalizing data. Since z-score values have zero mean, we can hide their value by adding random numbers from a distribution with zero mean and predefined standard deviation. As a result, users will all make computations with their z-scores instead of their actual ratings. To improve privacy level, we propose to hide unrated items of users, too. Users fill *f* (related with user's density) percent of their unrated items with random numbers having the same perturbation properties as employed for z-score disguising. Since having rating for an item shows that user has purchased this item, to hide which items are really purchased by any user, users fill *f* percent of their unrated items with random numbers.

The flow of data through the architecture is as follows:

At first, original rating profiles are masked. The decentralized protocol for data disguising is presented in the next section. The matrix generated from data disguising step is fed into the private trust module then. To realize the private trust module, at first, the trust is formalized to adapt to calculation with respect to z-scores. This is followed by disguising z-scores with randomness values. In our approach, users will apply the protocol described in following section to disguise their private vector during the process of trust estimation and computation. When users finish calculation of z-scores and data disguising, we can compute the trust between them. Within this framework, we have adopted Neil Lathia's [18] trust formalization for calculating interpersonal trust. The process of calculation is done in a decentralized fashion on each user's side. At the end of this step, trust values are returned and stored in the trust network. After this step, we use the private rating predictor module to produce final predictions, which generates the user-item ratings matrix, as the output of the recommender. To do so, we have adopted the *PCC* [28] along with the interpersonal trust values between users from previous step and generate *referrals*. Since users get normalized values from this output, the results are de-normalized. These steps are explained in more detail in the following section.

3.2 Data Disguising and Private Trust Computation

Once any user *a*, requests a recommendation from social recommender system, *a* need to disguise his/her rating vector to protect his/her own privacy. Therefore *a* firstly normalizes his/her rating vector and add randomness to normalized data. In our scheme users follow below protocols and processes for perturbing their data, estimating private trust, and producing private predictions.

3.2.1 Data Disguising Protocol

The procedure for data disguising is as follows:

1. Each user computes their ratings' z-score values.

2. Each user u selects a β value and then they uniformly randomly select standard deviation of the random numbers (σ_u) over the range [0, β]. They also compute number of rated items (num_{rat}) and number of unrated items (num_{unrat}) in their rating vectors.

3. Each user computes her density value d and she selects a random integer value f showing the percentage of unrated items to be filled between 0 and a number associated with d such as $d/2$, d, or $2d$. Each user u randomly selects f percent of their unrated items.

4. Users can utilize uniform distribution or Gaussian distribution to generate random numbers. To select the distribution, users decide a θ value over the range [0, 1] and they uniformly randomly select a random number r_u over the same range with θ.

    a. If $r_u \leq \theta$, the users generate random numbers having *uniform distribution* having interval [$-\delta$, δ], δ can be 1, 2, 3 or 4.

    b. Otherwise, they use *Gaussian distribution* with zero mean and standard deviation σ_u.

5. After selecting distribution, each user generates ($num_{rat} + num_{unrat} * \frac{f}{100}$) random numbers having zero mean and σ_u. To disguise rated items, each user add num_{rat} of random numbers to rated items' z-score values and they fill randomly selected $num_{unrat} * \frac{f}{100}$ unrated items with other random numbers.

6. Each user saves their masked z-score vectors.

3.2.2 Private Trust Estimation

As mentioned private trust module allows us to generate private trust values. Assume there are two users; u_a and u_b. We formalize the trust between them as follows:

$$z(u,i) = \frac{R_{u,i} - \overline{R_u}}{\sigma_u} \qquad (1)$$

Where $R_{u,i}$ is the true rating of user u on item i, $\overline{R_u}$ is the mean rating of user u, σ_u is the standard deviation of user u's ratings and $z(u,i)$ is the z-score value of user u on item i.

$$z'(u,i) = z(u,i) + r_{u,i} \quad (2)$$

Where $r_{u,i}$ is the random number generated by u to disguise z -score of item i and $z'(u,i)$ is the masked value of $z(u,i)$. When users finish calculating z-scores and data disguising, they compute trust among other users using Eq. 3:

$$T'(u_a,u_b) = 1 - \frac{\sum_{i=1}^{n} z'_{u_a,i_i} - z'_{u_b,i_i}}{z'_{max} * n} \quad (3)$$

This equation is an adapted formalization of trust proposed by Lathia et al. [18], which is based upon difference of a user's rating and its recommender's rating to their common item(s).

Here $T'(u_a,u_b)$ is the estimated (private) trust between respective users; u_a and u_b, z'_{u_a,i_i} is the masked z-score of user u_a for item i_i and z'_{max} is maximum masked z-score.

When u_a and u_b computes trust, they follow the steps below:

1. u_a and u_b decide which half they will operate on.
2. u_a and u_b send the parts that they will not operate on to each other.
3. When users receive related part of other user's vector, they compute sub-result of trust using Eq. 3.
4. Each user sends her sub-result to other user.
5. They compute trust value between each other by summing up sub-results.

3.2.3 Private Recommendation Prediction Process

To produce recommendations, Eq. 4 can be used. Since, z-scores are used instead of actual ratings in our scheme, when users finished computing trust; they use Eq. 4 to produce referrals, as follows;

$$p(a,i) = \frac{\sum_{b \in N_{(a,i)}} z'_{b,i} * T'(a,b)}{\sum_{b \in N_{(a,i)}} T'(a,b)} \quad (4)$$

Users get a normalized rating value when they use Eq. 4. To obtain actual rating value, users need to de-normalize result of Eq. 4 by using Eq. 5.

$$P(a,i) = \overline{R_u} + \sigma_u * p \quad (5)$$

Where $P(a,i)$ is the denormalized prediction for user a and item i, $\overline{R_u}$ is the mean rating for user u, σ_u is the standard deviation of user u's ratings and p is the referral value from previous step.

3.3 Defining and Inferring Optimal Privacy Set

It is accepted that privacy and accuracy are conflicting goals in the context of personalization and Collaborative filtering recommenders [12]. This conflict becomes more imminent in the presence of trust. Utilization of interpersonal trust aims at increasing [25], or maintaining the overall accuracy [22]. Trust metrics along with other factors such as neighbors' list size at each step of trust estimation increase or maintain the accuracy of predictions. This is while increasing the amount of perturbations leads to further information loss. To protect the private data, the level of perturbation is vital. If the amount is too low, the masked data still discloses considerable amounts of information; if it is too high, accuracy will be very low [12]. If we take into account the configurations that affect the privacy mechanism at one hand, and take into account the configurations affecting trust in another hand, we can argue that an optimal setting can be defined where privacy and accuracy can be both maintained at the same time. From the perspective of achieving an optimal result, problem space can be seen as an optimization design space. Within this design space we have *j* real parameters corresponding to trust mechanism configurations, while we have *k* different criteria corresponding to privacy mechanism configurations. In this space we take privacy enhancing mechanism as a function *p*, which generates privacy configurations set. For example, as we have used perturbation to protect private data, these operators become the distributions we have utilized for adding perturbations to the user rating profiles. We refer to this set as a *Privacy Configuration Set (PCS):*

$$\prod_{i\in I} pcs_i = \{p: I \to \bigcup_{i\in I} pcs_i \mid (\forall i)(p(i) \in pcs_i)\}$$

Since in theory, we can have an infinite number of parameters, we consider that at each time only *j* parameters are taken into account:

$$\Phi_j: \prod_{i\in I} pcs_i \to pcs_j$$

$$\Phi_j(p) = \{(pcs_1, pcs_2, \dots, pcs_j) | pcs_1 \in \Phi \wedge pcs_1 \in \Phi \dots \wedge pcs_j \in \Phi\}$$

In a similar fashion, trust enhancing mechanism, can be taken as a function *t*, which generates trust configurations set. For example, configurations that our trust-aware recommender uses to enable social network mediated trust inference are trust metric and the size of the trust lists, at each step. We refer to this set as a *Trust Configuration Set (TCS)*:

$$\prod_{i\in I} tcs_i = \{t: I \to \bigcup_{i\in I} tcs_i \mid (\forall i)(t(i) \in tcs_i)\}$$

Since in theory, we can have an infinite number of parameters, we consider that at each time only *k* parameters are taken into account:

$$\Psi_k : \prod_{i \in I} tcs_i \rightarrow \ tcs_k$$

$$\Psi_k(t) = \{(tcs_1, tcs_2, \dots, tcs_k) | tcs_1 \in \Psi \wedge tcs_1 \in \Psi \dots \wedge tcs_k \in \Psi\}$$

Since we study and analyze configurations from both mechanisms at once, then we need to make a joint set containing members from both sets. As a result we define an ordered set composed from Cartesian product of all privacy configuration sets (PCS), and trust configuration sets (TCS), as follows:

$$\psi = \Phi \times \Psi$$

$$\psi = \{(\ pcs_1, \dots, pcs_j, tcs_1, \dots, tcs_k) \mid (pcs_j) \in \Phi_j \wedge (tcs_k) \in \Psi_k\}$$

As the goal is achieving acceptable accuracy and respective privacy at the same time then optimization problem becomes multi-objective. As a result, problem of achieving a trade-off between accuracy and privacy in the current context becomes a *Pareto optimization problem*.

Taking into account this fact, we define an Optimal Privacy Set (OPS) as follows:

Definition. *Let ψ be the set of all possible joint configurations. There exists a set ψ_i, in which all possible joint privacy and trust configurations achieve a decent privacy and accuracy at the same time, in comparison to ψ_i^*, which is the other possible joint configurations. Such set exhibits Pareto optimality. We refer to this set as an Optimal Privacy Set (OPS):*

$$\psi_i \succcurlyeq \psi_i^*$$

In other words, among all possible configurations we can always find at least one setting that can either maintain or improve privacy, in the face of accuracy loss. To find such set, following heuristic can be adopted:

Heuristic. *To infer OPS, following heuristic is used:*

1. *Perturbing the overall user data using different PCS settings;*

2. *Observing the framework under variations of TCS;*

3. *Perturbing the sparse user data with PCS inferred from step 2 allows for inferring OPS and finalizing the Pareto optimal setting.*

In the evaluation section, through experiment we show how such set is inferred and justified as the optimal result, which respects the tradeoff we are trying to achieve.

4 Recommendation Framework Evaluation

To evaluate our framework, we have conducted two sets of experiments: First set demonstrates the effect of insertion of random data on accuracy of predictions generated as output of the recommendation system. The second set of experiments demonstrates how filling unrated items with varying f values affect the overall accuracy of recommender system. At the end, we define and infer the optimal privacy set with respect to experiment results. To measure the accuracy of recommendation system, we have utilized MAE (Mean Absolute Error) as respective metric. MAE measures the average absolute difference between predicted rating score made for a specific user and the user's actual rating [30]. For these experiments we have used public MovieLens dataset [33]. This dataset contains 943 user rating profiles, with more than 100000 rating values. Rating values are on a 5 point scale. For the first experiment part we have divided the profiles into 80% of data for training purpose and 20% for testing purposes. For the second part we have used 60% of data for training purpose and 40% for testing purpose.

4.1 Accuracy under overall masked user data

We have masked users' profiles with different random numbers having Gaussian distribution and Uniform distribution to show effects of distributions on accuracy. To setup the experiment we change two set of parameters: parameters that affect the data disguising operations, and parameters that affect the overall private trust computations. With respect to former, we have tried changing β and δ values while with respect to latter we have tried changing the t (trust metric value), n (neighbor's lists size). To compare the results under masked data with results without masked data, MAE for variations of t and n are presented in Fig. 2.

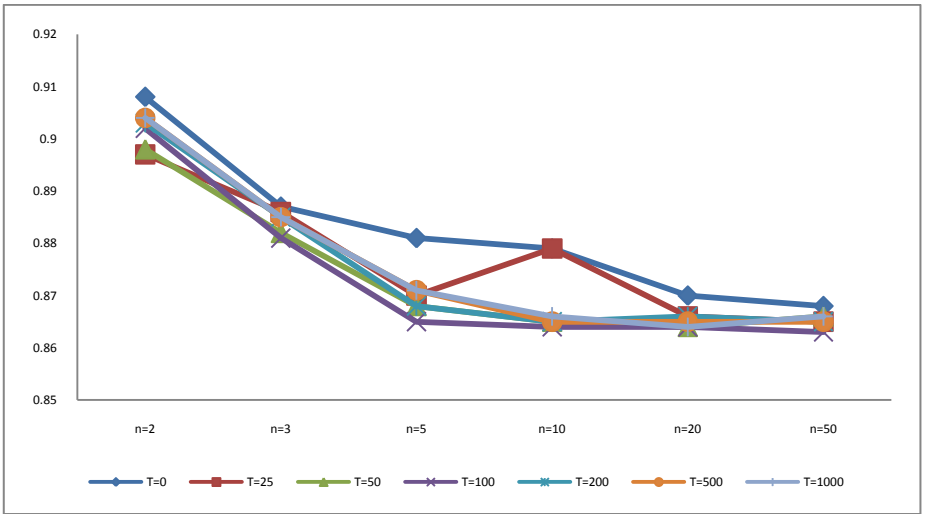
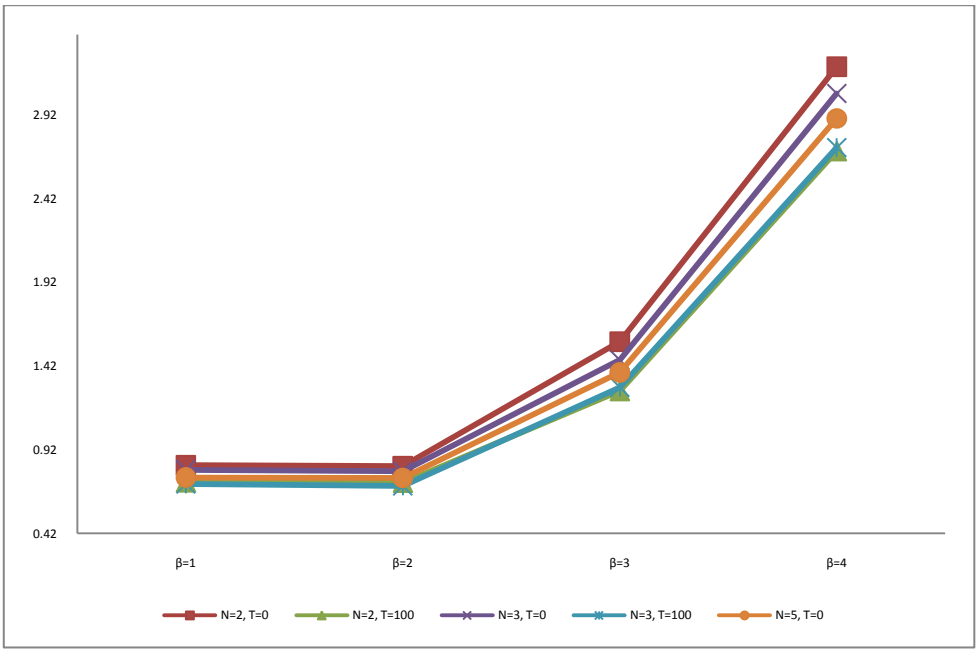


Fig. 2. MAE of recommendation framework, without adding any perturbations [31].



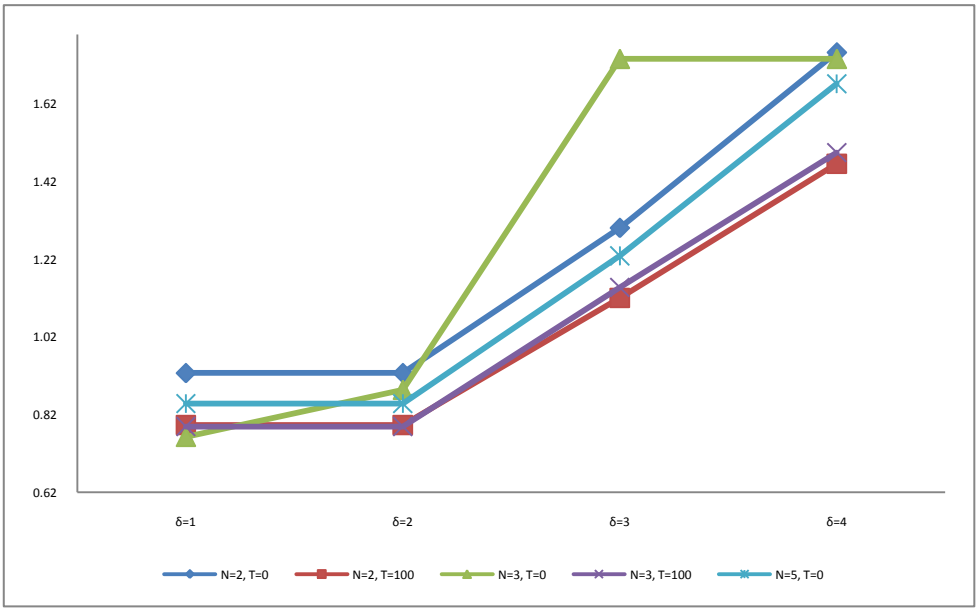


Fig. 3. Effects of adding perturbations on MAE, having Gaussian distribution (left), and having Uniform distribution (right), to user data.

Results of experiments with perturbation on MAE are depicted in Fig.3. Fig 3 plots the effects of random data on MAE with Gaussian distributions (Fig 3.a), and with Uniform distributions (Fig 3.b). In the case of Gaussian distribution we have selected {0.5, 1, 2, 4} for β as respective values, and for uniform distributions we have selected {1, 2, 3, 4} for δ as respective values. With respect to trust metric *t* values are selected from {0, 100} and for neighbors' lists we have tested with list sizes of 2, 3, and 5. In both plots, horizontal axes depict the possible intervals for different distributions of β and δ. Results of MAE experiments clearly state that if we utilize Gaussian distribution for random numbers the higher the β values, the better privacy is achieved and this is due to increasing randomness. We can witness the tradeoff here: The higher the β values the more accuracy we will lose. Results of Uniform distributions also confirm this observation. If we utilize Gaussian distribution for random numbers the higher the δ values, the better privacy is achieved, and the higher the δ values the more accuracy loss we have. With respect to *n*, we can observe in MAE results that are neither too high and nor too low values for *n* neighbor's list size can give us decent results. This is also the case for *t* where lowest value *(t =0)*, doesn't give uniform and consistent result, while highest value for *t (t =100)* yields more reasonable MAE results. Overall observation of MAE states that Gaussian distribution seems to be better than uniform distribution for accuracy but they are both useful with selected appropriate β and δ values.

4.2 Accuracy under sparse masked user data

To show effects of filling unrated items with random numbers, we have performed experiments with varying *f* values.

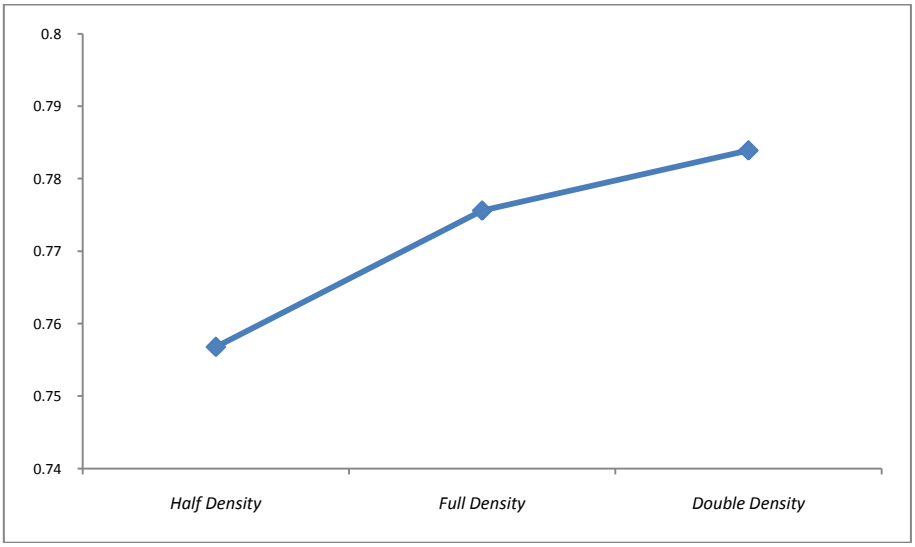


Fig. 4. Filling unrated items with random data having Gaussian distribution with respect to f.

In these experiments random numbers having Gaussian distribution with zero mean and standard deviation β=1 were used. We selected f values from the intervals [0, d/2], [0, d] and [0, 2d]. With respect to f we have depicted resulting MAE in Fig.4.

We can observe from the results that filling unrated items with random numbers provides better privacy, but it decreases accuracy as expected. Also when we increase possible f's interval, we achieve higher privacy level.

4.3 Analyzing trading-off between privacy and accuracy

Now that the base experiments were presented, we can take into account the privacy and accuracy metrics of current system to define and derive an optimal setting where system exhibits a transparent outcome.

Considering the parameters from both privacy and trust mechanisms, Φ and Ψ are defined as follows:

$$\Phi = \{(\beta, \delta, f) | \, \beta \in [1,5], \delta \in [1,5], f \in [0,2d]\},$$

$$\Psi = \{(n, t) | \, n \in [1,5], t \in [0,100]\}$$

Adopting the formalization introduced earlier, an optimum configuration is a joint setting of ψ= *(β, δ, f, n, t)* through which we maintain accuracy and privacy at the same time.

As a matter of fact, following the heuristic presented earlier;

1. First we perturb the overall user data using Gaussian and Uniform distributions *(δ, β)*, by comparing the results of MAE of framework under masked data (Fig.3), we can observe that set of *(δ, β) = (1,1)* yield best results as it exhibits the minimal privacy loss. As a result we fix *β=1, δ=1* for the next step.

2. In this step we observe the framework under variations of (n, t): With respect to this step, by comparing the results from MAE of framework under masked data (Fig.3), we observe that set of *(n, t) = (3,100)*, while being fixed on *(δ, β) = (1,1)*, yields reasonable accuracy, while privacy is maintained. So we fix the current set to (*δ, β, n, t) = (1,1,3,100)*.

3. In the final step we perturb the sparse user data with (*δ, β, n, t)* inferred from previous step for fine-tuning the privacy. To do so we utilize of different intervals of f with the system being fixed on (*δ, β, n, t)* configuration from previous step. Through observation of consistent accuracy of different f intervals, we can fine-tune the configuration from previous step and infer an optimum privacy configuration. Taking into account the results (Fig. 4), we observe consistent increase in intervals of f which finalizes the choice of *n, t, δ, β* and finalizes the results in ordered set of *n=3, t=100, δ=1, β=1* and f = [0, d] supporting both accurate and private recommendations:

$$\psi\ (\delta, \beta, n, t, f) = (1,1,3,100, [0, d])$$

Considering the existing range of ψ configurations, experiment showed that Pareto optimality holds.

These results were inferred with framework under masked user data. To make sure that optimum result maintains the Pareto optimality effect, we compare the MAE results of non-masked framework (Fig.2) with framework under masked results (Fig.3). In our work we inferred the optimum values for $\beta=1$, $n=3$ and $t=100$ and for these parameters $MAE= 0.7994$, while for similar parameters without adding perturbations we achieve $MAE=0.881$, which clearly shows that Pareto optimality holds, while it also shows that we have increased the privacy of the base framework with our architecture.

Further observation shows that our MAE results are still less than results of MAE without adding perturbations. According to (Fig.2), we achieve the best results with $MAE= 0.863$ for $(n,t)=(50,100)$ and this value is still greater than our optimum value. This observation also states that our result with proposed framework still shows better accuracy than the base framework.

5 Conclusion and Future Work

In this paper we proposed a framework for addressing the problem of privacy in trust recommenders. To overcome this obstacle, first we introduce a framework for enabling privacy-preserving trust-aware recommendation generation. After introduction of architecture, its building blocks and protocols, we pointed out the conflicting goals of privacy and accuracy. Within this context, we showed that a Pareto set can be always be found which can make a tradeoff between these conflicting aims and we presented a heuristic that experimentally infers this set. Through experimentation with predictive accuracy of private trust recommender system, we showed that we can infer such setting that holds even when trust recommender is not under privacy measures. We also showed that privacy increases under proposed framework, while even optimal privacy of our framework is better than the best performance of base framework in its best configurations. As a result privacy can be introduced in trust recommenders and can be optimized to avoid private data loss and at the same time produce accurate recommendations.

As future work, we plan to strengthen our framework against shilling attacks. We will investigate how to extend our scheme when data is collected by a central server

Acknowledgement. This work was partially supported by grant number 621-2007-6565 funded by Swedish Research Council and Grant 108E221 from TUBITAK.

References

1. Resnick, P., Iacovou, N., Suchak, M., Bergstorm, P., and Riedl, J. 1994. Group- Lens: An open architecture for collaborative filtering of netnews. In Proceedings of the ACM 1994 Conference on Computer-Supported Cooperative Work. ACM, Chapel Hill, NC, USA, 175–186.
2. Goldberg, D., Nichols, D., Oki, B., and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. Communications of the ACM 35, 12, 61–70.
3. L. F. Cranor. 'I didn't buy it for myself' privacy and e-commerce personalization. In Proceedings of the ACM Workshop on Privacy in the Electronic Society, pages 111-117, 2003.
4. J. Canny. Collaborative filtering with privacy. In Proceedings of the IEEE Symposium on Security and Privacy, pages 45-57, 2002.
5. J. Canny. Collaborative filtering with privacy via factor analysis. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 238-245, Tampere, Finland, 2002.
6. W. Du and H. Polat. Privacy-preserving collaborative filtering. International Journal of Electronic Commerce, 9(4):9-36, 2005.
7. C. Kaleli and H. Polat. Providing private recommendations using naive Bayesian classifier. Advances in Intelligent Web Mastering, 43:515-522, 2007.
8. C. Kaleli and H. Polat. P2P collaborative filtering with privacy. Turkish Journal of Electric Electrical Engineering and Computer Sciences, 8(1):101-116, 2010.
9. C. Kaleli and H. Polat. Providing private recommendations on personal social networks. Advances in Soft Computing, 67:117-125, 2010.
10. R. Parameswaran. A robust data obfuscation approach for privacy-preserving collaborative filtering. PhD thesis, Georgia Institute of Technology, 2006.
11. Y. Eytani, T. Kuflik, S. Berkovsky, P. Busetta and F. Ricci. Collaborative filtering over distributed environment. In Workshop on Decentralized, Agent-based and social Approaches to User Modeling, in conjunction with the 10th International Conference on User Modeling, Edinburg, UK, 2005.
12. C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in WWW '05: Proceedings of the 14th international conference on World Wide Web. New York, NY, USA: ACM, 2005, pp. 22–32.
13. J. A. Golbeck, "Filmtrust: movie recommendations from semantic web-based social networks," in Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE, Department of Computer Science, University of Maryland, 2006, pp. 1314– 1315.
14. J. Golbeck, "Trust and nuanced profile similarity in online social networks," MINDSWAP Technical Report TR-MS1284, University of Maryland, College Park, Tech. Rep., 2007.
15. Massa, P., & Avesani, P. Trust-Aware Collaborative filtering for Recommender Systems. In On the Move to Meaningful Internet Systems: CoopIS, DOA, and ODBASE (pp. 492-508), 2004.
16. Massa, P., & Avesani, P. Trust Metrics in Recommender Systems. In Computing with Social Trust (pp. 259-285), 2009.
17. J. O'Donovan and B. Smyth, "Trust in recommender systems," in IUI '05: Proceedings of the 10th international conference on intelligent user interfaces. New York, NY, USA: ACM, 2005, pp. 167–174.
18. N. Lathia, S. Hailes, and L. Capra, "Trust-based collaborative filtering," in IFIPTM 2008: Joint iTrust and PST Conferences on Privacy, Trust management and Security, Department of Computer Science, University College London, London, UK, 2008, p. 14.
19. Zarghami, A., Fazeli, S., Dokoohaki, N., & Matskin, M. (2009). Social Trust-Aware Recommendation System: A T-Index Approach. In Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on (Vol. 3, pp. 85-90). IEEE Computer Society. doi: 10.1109/WI-IAT.2009.237.
20. Squicciarini, a., Bertino, E., Ferrari, E., Paci, F., & Thuraisingham, B. (2007). PP-trust-X. ACM Transactions on Information and System Security, 10(3), 12-es. doi: 10.1145/1266977.1266981.
21. Singh, A., & Liu, L. (2003). TrustMe: anonymous management of trust relationships in decentralized P2P systems. In Proc. Third Int'l IEEE Conf. on Peer-to-Peer Computing.
22. Lam, S. K., Frankowski, D., & Riedl, J. (n.d.). Do You Trust Your Recommendations? An Exploration of Security and Privacy Issues in Recommender Systems. The New York Times, 1-15.
23. Mobasher, B., Burke, R., Bhaumik, R., & Williams, C. (2007). Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. ACM Trans. Internet Technol., 7(4). ACM. doi: 10.1145/1278366.1278372.

24. Lam, S. K., & Riedl, J. (2004). Shilling recommender systems for fun and profit (pp. 393-402). New York, NY, USA: ACM. doi: 10.1145/988672.988726.
25. F. Zhang, "Average shilling attack against trust-based recommender systems," International Conference on Information Management, Innovation Management and Industrial Engineering, vol. 4, pp. 588-591, 2009.
26. Massa, P., & Avesani, P. (2005). Controversial users demand local trust metrics: An experimental study on epinions.com community. In proceedings of the national conference on artificial intelligence (Vol. 20, p. 121). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.
27. J. E. Hirsch, "An index to quantify an individual's scientific research output," PNAS, vol. 102, no. 46, pp. 16 569–16 572, November 2005.
28. John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the l4h Conference on Uncertainty in Artificial Intelligence (UAI-98), pages 43- 52, San Francisco, July 24-26 1998.
29. N. Dokoohaki and M. Matskin, "Effective design of trust ontologies for improvement in the structure of socio-semantic trust networks," International Journal On Advances in Intelligent Systems, vol. 1, no. 1942-2679, pp. 23-42, 2008.
30. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," ACM Trans. Inf. Syst., vol. 22, no. 1, pp. 5–53, 2004.
31. S. Fazeli, A. Zarghami, N. Dokoohaki, and M. Matskin, "Mechanizing Social Trust-Aware Recommenders with T-index Augmented Trustworthiness," proceedings of the 7th International Conference on Trust, Privacy & Security in Digital Business (Trustbus 2010), in conjunction with the 21st International Conference on Database and Expert Systems Applications (DEXA 2010), Bilbao, Spain: 2010.
32. S. Fazeli, A. Zarghami, N. Dokoohaki, and M. Matskin, "Elevating Prediction Accuracy in Trust-aware Collaborative filtering Recommenders through T-index Metric and TopTrustee lists," the Journal of Emerging Technologies in Web Intelligence (JETWI), 2010.
33. J. Riedl, J. Konstan - Movielens dataset, 1998.