Differential Privacy in Online Dating Recommendation Systems

Teresa Anna Steiner

DTU Compute,
Technical University of Denmark, DK-2800 Kongens Lyngby, Denmark
terst@dtu.dk

Abstract. By their very nature, recommendation systems that are based on the analysis of personal data are prone to leak information about personal preferences. In online dating, that data might be highly personal. The goal of this work is to analyse, for different online dating recommendation systems from the literature, if differential privacy can be used to hide *individual* connections (for example, an expression of interest) in the data set from any other user on the platform - or an adversary that has access to the information of one or multiple users. We investigate two recommendation systems from the literature on their potential to be modified to satisfy differential privacy, in the sense that individual connections are hidden from anyone else on the platform. For Social Collab by Cai et al. we show that this is impossible, while for RECON by Pizzato et al. we give an algorithm that theoretically promises a good trade-off between accuracy and privacy. Further, we consider the problem of *stochastic matching*, which is used as the basis for some other recommendation systems. Here we show the possibility of a good accuracy and privacy trade-off under edge-differential privacy.

1 Introduction

By their very nature, recommendation systems that are based on the analysis of personal data are prone to leak information about personal preferences. Calandrino et al. [3] showed that given little auxiliary information even a passive adversary can infer other user's individual transactions for many user-to-item recommendation systems from the recommendations given to them by the system. In online dating that data might be highly personal. The goal of this work is to analyse, for different online dating recommendation systems from the literature, if differential privacy can be used to hide *individual* connections (for example, the existence of a dialogue, or an expression of interest) in the data set from any other user on the platform - or an adversary that has access to the information of one or multiple users.

Our contribution is summarized as follows. We investigate two recommendation systems from the literature on their potential to be modified to satisfy differential privacy, in the sense that individual connections are hidden from anyone else on the platform. For Social Collab [2] we show that this is impossible, while for RECON [21] we give an algorithm that theoretically promises a good accuracy and privacy trade-off. Further, we consider the problem of *stochastic matching*, which is used as the basis for some other recommendation systems [22, 4]. Here we show the possibility of a good accuracy and privacy trade-off under edge-differential privacy, though the running time of the algorithm is exponential in the size of the input graph.

Related Work. The definition of differential privacy is due to Dwork et al. [6]. It gives strong privacy guarantees and has been broadly researched in the contexts of data analysis and machine learning. For surveys on differential privacy in general and its applications in machine learning specifically see for example [7, 14, 25].

For a survey on privacy aspects of recommender systems see [9]. Differential privacy in recommendation systems has been widely researched in applications where items (goods at an auction, movies, etc.) are recommended to users [18, 12, 26, 10, 8, 16, 23]. A common feature of all differentially private methods for preserving privacy in recommendation systems is that the goal is to hide individual connections (purchases, ratings, etc.), which is our goal as well. There are two main differences in this work:

1. The recommendation systems themselves are different: all recommendation systems considered in this work are *reciprocal*, which means they take the preferences of both parties into account, that is, the taste of the recommended user and the user that is recommended to are considered. For online dating reciprocal recommendation systems have been shown to widely outperform non-reciprocal ones in practice [20].
2. The data that needs to be protected is different. While in most other applications, the privacy of the items is unimportant, here, everyone's privacy matters equally.

Further, Machanavajjhala et al. [17] studied social recommendation systems under differential privacy. The recommendation systems in their work are based on the assumption that it is much more likely that a user will form a connection if any of their friends formed the same connection. They concluded that a good accuracy and privacy trade-off is not possible for those systems. None of the recommendation systems considered in this work use direct mutual connections.

The stochastic matching problem is equivalent to the maximum matching problem. Matching and allocation problems have only been studied more recently in the differential privacy literature [13, 1, 15]. Notably, Hsu et al. [13] give an infeasibility result for differentially privately matching goods to people. Their counter-example heavily relies on having multiple copies of the same good (and as such, the same weight from a person to copies of the same good). Consequently, Hsu et al. [13] and Kannan et al. [15] use relaxed versions of differential privacy. In contrast, we assume that the weights are independent probabilities. Anandan and Clifton [1] focus on constructing a data oblivious algorithm and then show how to output the value of the minimum matching in a differentially private way.

Outline. This paper is structured as follows. In the preliminaries (Section 2) we define differential privacy, collect some basic results, and introduce the recommendation systems we analyse in this work. In Section 3.1 we show by a counter-example that a sensible privacy and accuracy trade-off is impossible for the Social Collab recommendation system by Cai et al. [2]. In Section 3.2 we give a differentially private algorithm for RECON based on the Laplace mechanism and give arguments as to why this seems to promise a good trade-off. In Section 3.3 we consider the stochastic matching problem, and give a differentially private algorithm with a good accuracy trade-off, but which is inefficient. The algorithm achieves essentially the same as the exponential mechanism by McSherry and Talwar [19] on the set of all matchings, but we provide a direct analysis for our specific application.

2 Preliminiaries

In this section we define differential privacy, collect some basic results, and introduce the recommendation systems we adapt in this work.

2.1 Differential Privacy

The definition of differential privacy is due to Dwork and McSherry [6]. The basic idea is to find a randomized algorithm with the property that the output distributions are similar if the input data sets differ in a single entry. This is generally achieved by adding noise at the cost of accuracy.

For the formal definition, we need the notion of *neighbouring data sets.*

Definition 1. *Let $\mathcal{D}$ be a universe of data sets. Two data sets $x \in \mathcal{D}$ and $y \in \mathcal{D}$ are called* neighbouring *if they differ in at most one entry (what this means exactly will depend on $\mathcal{D}$). We will write $x \sim y$.*

Now, the differential privacy property says that the output distributions of a randomized mechanism have to be close for neighbouring data sets.

Definition 2. *A mechanism $\mathcal{M}$ is called* (ϵ, δ)-differentially private *if for every S in the range of $\mathcal{M}$ and two neighbouring data sets x and y it holds that*

$$P\left(\mathcal{M}\left(x\right) \in S\right) \leq \exp\left(\epsilon\right) P(\mathcal{M}\left(y\right) \in S) + \delta. \tag{1}$$

For small values of ϵ and δ this means that the output distributions are similar if the data sets differ in a single entry. If $\delta = 0$, the function is called ϵ-differentially private, and the smallest parameter ϵ for which (1) is true is referred to as the “privacy loss”.

The results presented in this work use the Laplace mechanism by Dwork et al. [6] as a basic tool. We need the notion of *global sensitivity*, which is a measure for the maximum output difference of a function evaluated on two neighbouring data sets.

Definition 3. *The* global sensitivity *of a function $f : \mathcal{D} \to \mathbb{R}^d$ is defined as*

$$\Delta f = \max_{x \sim y} \|f(x) - f(y)\|_1. \tag{2}$$

Next, we define the Laplace distribution.

Definition 4 (Laplace Distribution). *The Laplace distribution* $\mathrm{Lap}\,(\mu, b)$*, where μ is the mean and $b > 0$ is the scale parameter, is defined by the density function $f(x|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right)$. If we omit the first parameter we assume $\mu = 0$.*

The Laplace mechanism adds Laplace noise scaled with the global sensitivity to the output of a function.

Theorem 1 (Dwork et al. [6]). *For a function $f : \mathcal{D} \to \mathbb{R}^d$ let $(Y_1, \ldots, Y_d)$ be independent random variables drawn from the Laplace distribution with scale parameter $b = \frac{\Delta f}{\epsilon}$. Then the mechanism $\mathcal{M}(x) := f(x) + (Y_1, \ldots, Y_d)$ is ϵ-differentially private. This mechanism is called the* Laplace mechanism*.*

Further, we need two basic results about differential privacy. The first one is the post-processing rule, which states that any transformation of a differentially private output preserves the same (or better) privacy guarantees.

Lemma 1. *If $\mathcal{M}$ is differentially private, any mechanism $g \circ \mathcal{M} : x \mapsto g\left(\mathcal{M}(x)\right)$ is also differentially private, for any function g defined on the range of $\mathcal{M}$.*

Secondly, a basic composition theorem by Dwork et al. [5] states that the privacy loss of a composition of several differentially private functions is no more than the sum of the privacy losses of each individual function.

Theorem 2 (Dwork et al. [5]). *The composition of k differentially private mechanisms with parameters $(\epsilon_1, \delta_1), \ldots, (\epsilon_k, \delta_k)$ is (ϵ, δ)-differentially private with $\epsilon = \sum_{i=1}^{k} \epsilon_i$ and $\delta = \sum_{i=1}^{k} \delta_i$.*

When working with graphs, there are mulitple ways of defining neighbouring data sets [11], depending on whether the goal is to hide a person's presence in the data set (resulting in the definition of *node differential privacy*), or individual connections. The latter is formalized by the definition of *edge-differential privacy*, which is due to Hay et al. [11].

Definition 5 (Edge-Differential Privacy). *Let $\mathcal{D}$ be a set of graphs on a vertex set V with $|V| = n$. A mechanism is* edge-differentially private*, if it is differentially private over $\mathcal{D}$, when for $x = G_x = (V, E_x) \in \mathcal{D}$ and $y = G_y = (V, E_y) \in \mathcal{D}$ the neighbouring relation $x \sim y$ means there exists an edge e such that $E_x = E_y \backslash \{e\}$ or $E_y = E_x \backslash \{e\}$.*

In this work we will only consider variations of edge-differential privacy.

2.2 Online Dating Recommendation Systems

By a *recommendation system*, in general form, we mean any mechanism that uses data from an online dating platform, and gives an individual, ranked list of users as recommended connections to every user on the platform. When we talk about the recommendations given to one specific user, we call this user the *active user*. Often, the space of potential connections for one active user is not the entire data base, but some restricted subset based on for example location, gender, age, etc. The choices for recommendation systems considered in this work are mostly influenced by an extensive review by Pizzato et al. [20]. Additionally to summarizing and categorizing existing strategies, they point out two specific characteristics which show improved success rate of recommendations:

- Recommendation systems which are based on *implicit* preferences (i.e. based on existing matches and past behaviour on the platform) are more effective than *explicit* preferences (i.e. preferences the user states).
- Recommendation systems which take preferences of both the active user and the user we potentially want to recommend into account are more effective than considering preferences of the active user only.

Social Collab. In the model by Cai et al. [2] the data set is modelled as a directed graph, where nodes represent users and edges represent some form of “like”, for example liking a profile, or sending a message, or the positive reply to a message. They define two users to be *similar in attractiveness* if they were liked by at least one person in common, and *similar in taste* if they liked at least one person in common; see Figure 1 (a). A user r is a *predicted match* for the active user a if there exists

1. at least one user similar in taste to r that liked a and
2. at least one user similar in attractiveness to r that was liked by a.

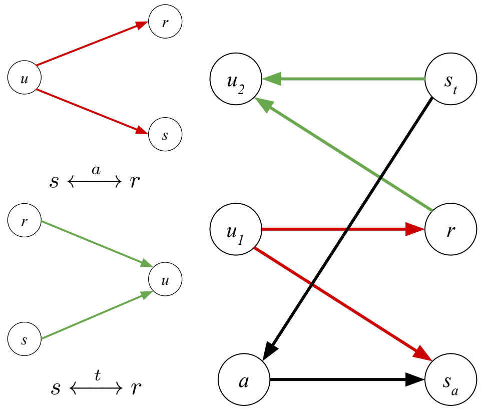
See Figure 1 (b).

All predicted matches for user a are ranked by the number of total users for which either of the two conditions above is true. The output is, for some N, the sorted list of the top N predicted matches.

RECON. The RECON system by Pizzato et al. [21] considers a model where users can like other users’ profiles. Each profile consists of several attributes. Based on the attributes of users that a liked before and b’s profile, they predict a score $C^+(a,b)$ of how likely it is that user a will like user b. The data set is modelled as a directed graph where users are nodes and an edge (u,v) means that user u liked user v. Let $\mathcal{N}(a)$ be the set of users that user a liked, and $d(a) = |\mathcal{N}(a)|$ the outdegree of a. Denote $\mathrm{Att}(b)$ the set of all attributes that user b possesses.

Definition 6. *The* positive compatibility $C^+(a,b)$ of a candidate b for user a *is defined as follows:*

$$C^+(a,b) = \frac{\sum_{u\in\mathcal{N}(a)} \sum_{t\in\mathrm{Att}(b)} \mathbb{1}\left(t \in \mathrm{Att}(u)\right)}{d(a)\,|\mathrm{Att}(b)|}.$$



(a) (b)

Fig. 1: Social Collab. (a) The users s and r are similar in attractiveness or taste, respectively. (b) Since s_t is similar in taste to r and s_t likes a, and s_a is similiar in attractiveness to r and is liked by a, user r is a predicted match for active user a.

As mentioned before, an important characteristic that has been shown to improve the success rate of recommendations is reciprocability. In their paper, Pizatto et al. [21] define a reciprocal version of RECON by using the harmonic mean between $C^+(a,b)$ and $C^+(b,a)$. The *positive compatibility between* a *and* b is defined as

$$C^+_{rec}(a,b) = C^+_{rec}(b,a) = \frac{2}{\frac{1}{C^+(a,b)} + \frac{1}{C^+(b,a)}}.$$

As such, a low compatibility score for one of either $C^+(a,b)$ or $C^+(b,a)$ will result in a low combined compatibility score. This means we will only recommend b to a if it is considered likely that b will like a back. Pizatto et al. [21] showed that the reciprocal version of RECON has a significantly higher success rate than the non-reciprocal one.

Stochastic Matching. Next, we will define the stochastic matching problem, which is not a recommendation system itself, but is used as a model for some of them. Instead of singling out one active user and ranking recommendations for this user, the idea here is to always recommend user b to user a exactly when we also recommend user a to user b. Instead of outputting a sorted list of recommendations for one user, we will output a list of total recommendations, and the goal is to have the same number of recommendations for each user. The motive of this strategy is, as argued by Pizzato and Silvestrini [20], to ensure that there is no difference made between popular and unpopular users. Receiving either too many or too few messages can cause frustration in a user.

Both the recommendation systems by Pizzato and Silvestrini [20] and Chen et al. [4] model the problem as a weighted, undirected graph $G = (V, E)$. The vertices in V represent the users, and for each edge $e = (u, v)$, the edge weight $0 \leq w(e) \leq 1$ is an estimate for the probability of a *successful match* between u and v. A successful match is defined differently in the two papers, but the common ground is that the recommendation leads to some sort of positive interaction between the users. For simplicity we assume $E = V \times V$ and set the weight of non-existing edges to zero. The methods used to estimate the edge weights differ for each system and will not be further discussed here.

The *weighted maximum matching problem* is defined as follows: given a weighted, undirected graph $G = (V, E)$ with a non-negative weight function w on E and a given number $N \geq 1$, the goal is to find a subset M of edges such that

1. any vertex is adjacent to at most N edges in M and
2. the value $w(M) := \sum_{e \in M} w(e)$ is maximized with respect to all subsets M satisfying condition 1.

We will call any subset of edges satisfying condition 1 a *matching* on G. Furthermore, we will restrict ourselves to the simplified case where $N = 1$.

In Pizzato and Silvestrini [20] and Chen et al. [4] the *stochastic matching problem* is formulated slightly differently: given the probabilities of any successful match, the goal is to maximize the expected number of successful matches, under the condition that each user receives at most N recommendations. Note that since the edge weights represent the probabilities of any match being successful, the problem is equivalent to the one formulated above.

3 Results

First, we have to define what privacy notion we aim to achieve. As mentioned in the preliminaries, one could aim to hide either the presence of a person in the data set, or individual connections. We focus on hiding individual connections. This has two reasons: it makes more practical sense, because the presence of someone on a dating platform can usually not be entirely hidden. Also, it is much easier to achieve, since an individual connection influences a recommendation less than the full data of a person.

Note that when recommending to a, we do not need to protect outgoing edges from a, since they only encode information about a's own preferences. Our aim is to hide a user's connections from anyone who has access to the recommendations of any other user on the platform. An adversary in this setting could be someone who creates fake profiles and gathers information from the recommendations given to those fake users. Formally, for a fixed active user a, we define two data sets $x = (V, E_x)$ and $y = (V, E_y)$ with $a \in V$ as neighbouring, $x \sim y$, if there exists an edge $e \neq (a, v)$ for any $v \in V$ such that $E_x \backslash \{e\} = E_y \backslash \{e\}$. This means two neighbouring data sets differ in an edge that is not an outgoing edge of a, because those are the edges we want to keep private when recommending to a.

We will use differential privacy based on this definition of neighbouring data sets for Social Collab and RECON.

3.1 Social Collab

We give a counter-example to show that it is not possible to directly modify Social Collab to satisfy differential privacy under the definition of neighbouring data sets described above.

Consider the (sub-)graph given by Figure 2. Assume we have an active user a and a subset of k possible candidates $r_1, \dots, r_k$. Everyone of the r_i, for $i = 1, \dots, k$, likes a user u_2, who is also liked by a set of users $s_{t1}, \dots, s_{tl}$. As such, all the users r_i and s_{tj} are similar in taste. Additionally, all users s_{tj}, for $j = 1, \dots, l$ like user a. Then, there is a user s_a who is liked by a, and a user u_1, which likes every r_i for $i = 1, \dots, k$. Now, if u_1 also likes s_a, all users r_i are predicted matches for a with weight $l+1$, since they are similar in taste to l users that liked a and similar in attractiveness to a user s_a who was liked by a. On the other hand, if the edge (u_1, s_a) does not exist, there is no user similar in attractiveness to any of the r_i which was liked by a, which means none of them is a predicted match.

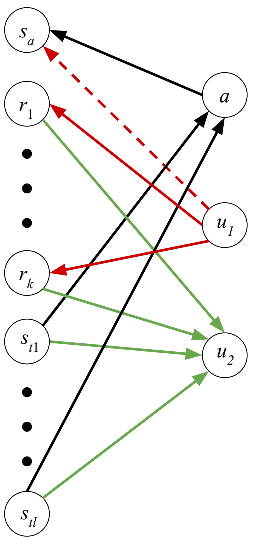


Fig. 2: This figure shows the counter-example for Social Collab. The dashed edge is present in a data set x, but not in its neighbouring data set y. Each s_{tj} is similar in taste to each of the r_i. In x, each r_i is similar in attractiveness to s_a.

Note that l and k can be arbitrarily large. This means that one edge can influence the rating of arbitrarily many users from very high, to not even being recommended at all. Thus, we cannot hope to preserve differential privacy here without destroying all information.

The example also demonstrates which kind of potential attacks we are trying to prevent: An adversary aware of the subgraph in Figure 2 can find out if the edge (u_1, s_a) exists by looking at recommendations for user a. As such, an adversary can try to build specific subgraphs using fake profiles to uncover additional information about the network.

3.2 RECON

The goal of this section is to find a way to privately output the vector consisting of $C^+_{rec}(a,u)$, for every u which is a candidate for a, to a under the privacy model defined in the beginning of Section 3. This means that two neighbouring data sets differ in an edge that is not an outgoing edge of the active user a.

First, notice that because differential privacy is immune to post-processing (Lemma 1), if we show how to make both $C^+(a,u)$ and $C^+(u,a)$ differentially private, their private versions can be used to compute $C^+_{rec}(a,u)$ while still preserving differential privacy.

Deleting any (u,v) for $u \neq a$ can not change $C^+(a,b)$ because that quantity depends only on outgoing edges of a and attributes of b. Thus, no noise has to be added to privately output $C^+(a,b)$. Consider now a as the active user. Denote U_a the set of candidate matches for a (e.g. users living close to a). The following Lemma shows that deleting or adding an edge (b,v) where $b \neq a$ can change only $C^+(b,a)$ by at most $\frac{1}{d(b)}$.

Lemma 2. *For any $b \in U_a$ with $d(b) > 1$, deleting or adding an outgoing edge of b can change $C^+(b,a)$ by at most $\frac{1}{d(b)}$.*

Proof. Fix a data set $x = G_x = (V, E_x)$. We consider neighbouring data sets y which differ from x in an outgoing edge from b. We will use x and y as subscripts for d, $\mathcal{N}(b)$ and C^+ to differentiate the value of the respective functions in the different data sets.

First, assume x and y differ in an edge (b,v) that is present in x but not in y. We have

$$\begin{aligned}
&\left|C^+_x(b,a) - C^+_y(b,a)\right| \\
&= \left| \frac{\sum_{u\in\mathcal{N}_x(b)} \sum_{t\in\mathrm{Att}(a)} \mathbb{1}\left(t\in\mathrm{Att}(u)\right)}{d_x(b)\left|\mathrm{Att}(a)\right|} - \frac{\sum_{u\in\mathcal{N}_x(b)\setminus\{v\}} \sum_{t\in\mathrm{Att}(a)} \mathbb{1}\left(t\in\mathrm{Att}(u)\right)}{(d_x(b)-1)\left|\mathrm{Att}(a)\right|} \right| \\
&= \left| \frac{\sum_{u\in\mathcal{N}_x(b)\setminus\{v\}} \sum_{t\in\mathrm{Att}(a)} \mathbb{1}\left(t\in\mathrm{Att}(u)\right)}{\left|\mathrm{Att}(a)\right|} \left(\frac{1}{d_x(b)} - \frac{1}{d_x(b)-1}\right) \right. \\
&\left. + \frac{\sum_{t\in\mathrm{Att}(a)} \mathbb{1}\left(t\in\mathrm{Att}(v)\right)}{d_x(b)\left|\mathrm{Att}(a)\right|} \right|.
\end{aligned}$$

Note that since

$$\left(\frac{1}{d_x(b)} - \frac{1}{d_x(b)-1}\right) = -\frac{1}{d_x(b)\left(d_x(b)-1\right)} < 0,$$

we have that the difference between $C_x^+(b,a)$ and $C_y^+(b,a)$ is bounded by

$$\max\left(\left|\frac{\sum_{u\in\mathcal{N}_x(b)\setminus\{v\}}\sum_{t\in\mathrm{Att}(a)}\mathbb{1}\left(t\in\mathrm{Att}(u)\right)}{|\mathrm{Att}(a)|\,d_x(b)\left(d_x(b)-1\right)}\right|,\left|\frac{\sum_{t\in\mathrm{Att}(a)}\mathbb{1}\left(t\in\mathrm{Att}(v)\right)}{d_x(b)\,|\mathrm{Att}(a)|}\right|\right).$$

Bounding each $\mathbb{1}\left(t\in\mathrm{Att}(u)\right)$ by 1 we get

$$\left|C_x^+(b,a) - C_y^+(b,a)\right| \leq \frac{1}{d_x(b)}.$$

In the other case where y has one edge (b,v) more than x, the analysis from above goes through with reversed roles of x and y and we get

$$\left|C_x^+(b,a) - C_y^+(b,a)\right| \leq \frac{1}{d_y(b)} < \frac{1}{d_x(b)}.$$

Assuming our data universe is such that there is a lower bound $T \leq d(u)$ for all potential matches u of a, this means that the global sensitivity of $\left(C^+(u,a)\right)_{u\in U_a}$ is bounded by $1/T$ - since one edge can influence one entry in the vector by at most $1/T$. As such, we can use the Laplace mechanism with scale $\frac{1}{T\epsilon}$ to privately output $\left(C^+(u,a)\right)_{u\in U_a}$. Then we can compute $\left(C_{rec}^+(a,u)\right)_{u\in U_a}$ privately by the post processing rule.

In the general case, we will not have a good lower bound on the outdegree of all potential candidates. In fact, in the worst case there might exist a user with outdegree zero, for example, a new user on the platform. Intuitively though, if a user u does not have a lot of outgoing edges, the information gained by considering $C^+(u,a)$ is not reliable anyway - there is not enough data to make conclusions about the user's taste.

We propose the following idea: we use the noisy $C_{rec}^+(a,u)$ only for elements $u \in U_a$ which have an outdegree above a certain threshold, and $C^+(a,u)$ for all others. This means that whenever u has sufficiently high degree, that is, sufficient activity on the platform, we use both u's and a's preferences in the recommendation. Otherwise, we will only consider a's preferences.

The difficulty with this approach is that the decision whether or not the degree is above a certain threshold can leak information. This means we first have to find a differentially private approximation of the outdegrees of all candidates u. The vector containing all outdegrees has a sensitivity of 1, since adding or removing an edge can change the outdegree of one node. In Algorithm 1, we use the Laplace mechanism for finding a differentially private degree sequence. Alternatively, one could use Hay et al.'s [11] modification for finding a differentially private degree sequence. We stick to the Laplace mechanism for simplicity.

Our approach is now summarized as follows: We first add noise to make the vector of outdegrees differentially private. Then we use this vector to decide if the outdegree for any u is above a certain threshold or below. Then, depending on which case we are in, we either use the idea described above to output a noisy $C^+_{rec}(a,u)$, or output $C^+(a,u)$ directly. The details are shown in Algorithm 1.

The $(T-\alpha)$ in Algorithm 1 originates from the fact that if we know that the noisy degree is above a certain threshold, we do not have a guarantee that the *actual* degree is above that threshold; but we will show that with appropriate choice of α, with high probability, the noisy degree will be above $T-\alpha$. If we are in that case, adding noise scaled with $\frac{1}{T-\alpha}$ is sufficient for preserving privacy. With non-zero probability, though, the noisy outdegree will be above T, while the actual one is smaller than $T-\alpha$. In that case, we cannot guarantee that the privacy loss in our algorithm is less than ϵ. We prove $(\epsilon,\delta)-$differential privacy for $\delta > 0$. The details follow in the proof of Theorem 3.

Algorithm 1 Private RECON $(a, U_a, \epsilon, T, \alpha)$

```
for every $u \in U_a$ do
    Let $Y_1 \sim \text{Lap}\left(\frac{2}{\epsilon}\right)$ and $Y_2 \sim \text{Lap}\left(\frac{2}{(T-\alpha)\epsilon}\right)$
    if $d(u) + Y_1 > T$ then
        Compute $\tilde{C}^+(u,a) := C^+(u,a) + Y_2$ and use it to
        compute $c_u := \frac{2}{\frac{1}{C^+(a,u)} + \frac{1}{\tilde{C}^+(u,a)}}$
    else
        set $c_u := C^+(a,u)$
return $(c_u)_{u \in U_a}$
```

Theorem 3. *Algorithm 1 is (ϵ,δ)-differentially private for $\alpha > \frac{2(\log(|U_a|)-\log(\delta))}{\epsilon}$.*

Proof. First, note that providing (ϵ,δ)-differential privacy is equivalent to guaranteeing a privacy loss of at most ϵ with probability at least $1-\delta$. Since the vector $(d(u))_{u\in U_a}$ has sensitivity 1, adding independent noise with scale $2/\epsilon$ makes the output of the noisy vector $(\epsilon/2)$-differentially private. This means outputting which users are in the first or second case of the algorithm preserves $(\epsilon/2)$-differential privacy.

As argued before, outputting $C^+(a,u)$ for each user u in the second case can be done without further privacy loss. The interesting case to consider is first case, when the noisy degree is above threshold T.

Claim. With probability at least $1-\delta$, all u from the first case satisfy $d(u) > T-\alpha$.

If the claim is true, then with probability $1-\delta$, adding independent Laplace noise scaled with $\frac{2}{(T-\alpha)\epsilon}$ to $C^+(u,a)$ for each u in the first case will preserve $(\epsilon/2)$-differential privacy - since by Lemma 2, changing one edge can change $C^+(u,a)$

for at most one u by at most $1/(T-\alpha)$, thus the L_1 norm of the vector by at most $1/(T-\alpha)$. By the post-processing rule and the composition theorem (Lemma 1 and Theorem 2) we are done.

Proof (of Claim). : Let u be any node with $d(u) \leq T-\alpha$. We then have that

$$P\left(d(u)+Y_1 > T\right) \leq P(|Y_1| > \alpha) = \exp\left(-\frac{\alpha\epsilon}{2}\right). \tag{3}$$

By the union bound, the probability that any u with $d(u) \leq T-\alpha$ gets classified into the first case is at most $|U_a| \exp\left(-\frac{\alpha\epsilon}{2}\right) < \delta$ by choice of α.

This concludes the proof of the theorem.

To preserve privacy in practice, the parameter δ is usually recommended to be $o(1/n)$, where n is the data set size. Since our algorithm only operates on the set U_a, if we choose e.g. $\delta = \frac{1}{|U_a|^2}$, we satisfy this condition and can choose $\alpha = \frac{6\log(|U_a|)}{\epsilon}$. Clearly, if we choose α as above, we have to choose $T = \Omega\left(\frac{\log(|U_a|)}{\epsilon}\right)$. Often, we require δ to be smaller than the inverse of any polynomial of the data set size [7]. Note that if we choose e.g. $\delta = \frac{1}{|U_a|^{\log|U_a|}}$, we can choose $T = \Theta\left(\frac{\log^2(|U_a|)}{\epsilon}\right)$. Now, the higher we choose T, the less noise we have to add to high degree nodes. On the other hand, the higher we choose T, the fewer nodes will be classified as high degree nodes, which means we take the preferences of fewer candidates into account. As such, a good threshold T can only be empirically optimized given data - assuming that we cannot estimate mathematically how many profiles a user would have to like in order for us to make sensible predictions about their taste.

3.3 Stochastic Matching

For the stochastic matching problem, we give an exponential running time algorithm which is both accurate and private, showing the theoretical feasibility of a good trade-off. The definition of privacy we use is edge-differential privacy from Definition 5. Note that for the privacy definition to make sense in this application, we implicitly assume that the edge weight estimations are independent for each edge, which might not be true for all applications. In [24] we show that the simple idea of adding Laplace noise to the weight of each edge fails to give a good privacy and accuracy trade-off. The algorithm we present here is similar to the algorithm Report Noisy Max found in [7]. The output distribution of this algorithm is almost equivalent to the *exponential mechanism* by McSherry and Talwar [19], as shown in [7]. In our algorithm, we compute all feasible matchings, and add Laplace noise to the value of each matching. Then we choose the matching with maximum noisy value.

Theorem 4. *Algorithm 2 is ϵ-differentially private.*

Algorithm 2 Noisy Max Matching (G, w, ϵ)

```
for every possible matching $M$ in $G$ do
    compute its weight $w(M)$
    draw $Y \sim \mathrm{Lap}\,(1/\epsilon)$
    set $\tilde{w}(M) = w(M) + Y$
return argmax$(\tilde{w})$
```

Proof. We will sketch the proof by showing the properties used in Claim 3.9 in [7], which proves ϵ-differential privacy for Report Noisy Max, and then follow the steps of their proof.

Fix two neighbouring data bases x and y such that there exists an edge e with $w_x(e) \geq w_y(e)$. For each possible matching M of G, denote its weight in x by $w_x(M) = \sum_{e \in M} w_x(e)$. Note that, since the graph without weights is the same in both data sets, the set of possible matchings is independent of the data set.

Now, it is easy to see that the two properties used in the proof in [7] hold:

1. Monotonicity: For each possible matching M we have $w_x(M) \geq w_y(M)$.
2. Lipschitz Property: For each possible matching M we have $1 + w_y(M) \geq w_x(M)$.

We will use these properties later in the proof. Denote by M_G the set of all possible matchings on G and fix one matching $M_0 \in M_G$. Define the vector of independent Laplace variables $Y = (Y_M)_{M \in M_G}$, where $Y_M \sim \mathrm{Lap}\,(1/\epsilon)$. That is, the algorithm

chooses $\mathrm{argmax}_{M \in M_G}\,(w(M) + Y_M)$. Further, denote Y_{-M_0} the random vector of Y without the entry corresponding to M_0.

Now, fix a realization z_{-M_0} of Y_{-M_0}, that is, a vector where each coordinate is drawn from $\mathrm{Lap}\,(1/\epsilon)$. We will show the property from Definition 2 for $\delta = 0$ separately for each condition $(Y_{-M_0} = z_{-M_0})$. For simplicity, we write z_M for the coordinate corresponding to matching M in z_{-M_0}.

Denote $r := \max_{M \in M_G}\,(w_x(M) + z_M - w_x(M_0))$. Thus, we output M_0 on database x if and only if $Y_{M_0} > r$. By the Monotonicity and Lipschitz property above, we have, for all $M \neq M_0$,

$$1 + w_y(M_0) + r \geq w_x(M_0) + r \geq w_x(M) + z_M \geq w_y(M) + z_M.$$

This means that if $Y_{M_0} > 1 + r$, then M_0 is the output on data base y. We have

$$\begin{aligned} P\left(\text{Noisy Max Matching }(y, \epsilon) = M_0 \middle| \, Y_{-M_0} = z_{-M_0}\right) &\geq P(Y_{M_0} > 1 + r) \\ &= \frac{1}{2}\exp\left(-\epsilon(1 + r)\right) \\ &= \exp(-\epsilon) P(Y_{M_0} > r) \end{aligned}$$

$$= \exp(-\epsilon) P \left(\text{Noisy Max Matching } (x, \epsilon) = M_0 | \, Y_{-M_0} = z_{-M_0}\right),$$

where for the first and second equality we use Theorem 1 and the symmetry of the Laplace distribution.

Similarly, one can prove the other inequality, that is,

$$\begin{aligned} &P \left(\text{Noisy Max Matching } (y, \epsilon) = M_0 | \, Y_{-M_0} = z_{-M_0}\right) \\ \leq &\exp(\epsilon) P \left(\text{Noisy Max Matching } (x, \epsilon) = M_0 | \, Y_{-M_0} = z_{-M_0}\right). \end{aligned}$$

When we have that, we are done, since if the inequalities hold under each realization of Y_{-M_0}, they also hold for the marginal distributions $P\left(\text{Noisy Max Matching } (x, \epsilon) = M_0\right)$ and $P\left(\text{Noisy Max Matching } (y, \epsilon) = M_0\right)$.

Next, we will show that this algorithm actually provides a good trade-off. The result is comparable to the accuracy achieved by outputting only the value of the maximum matching using the Laplace mechanism: Since the global sensitivity of the value is at most 1 and by the properties of the Laplace distribution, the expected error for this is at most $\frac{1}{\epsilon}$. Even though outputting an actual matching instead of only the value seems to provide much more information, we achieve almost the same error guarantees.

Lemma 3. *The value of the matching output by Algorithm 2 differs from the optimal by at most α with probability at least $1 - 2\exp\left(-\frac{\epsilon\alpha}{2}\right)$. The expected difference is at most $\frac{2}{\epsilon}$.*

Proof. Let M denote the matching output by Algorithm 2 and M^* any maximum matching. Further, denote Y_M the random variable added to $w(M)$ and Y_{M^*} the random variable added to $w(M^*)$ in the algorithm. We have

$$w(M) + Y_M \geq w(M^*) + Y_{M^*}.$$

It follows

$$|w(M) - w(M^*)| = w(M^*) - w(M) \leq Y_M - Y_{M^*} \leq |Y_M - Y_{M^*}|.$$

Using this, we get

$$\begin{aligned} P\left(|w(M) - w(M^*)| \geq \alpha\right) &\leq P(|Y_{M^*} - Y_M| \geq \alpha) \\ &\leq P(2\max(|Y_{M^*}|, |Y_M|) \geq \alpha) \\ &\leq 2P(2|Y_M| \geq \alpha) \\ &= 2P\left(|Y_M| \geq \frac{\alpha}{2}\right) = 2\exp\left(-\frac{\epsilon\alpha}{2}\right) \end{aligned}$$

where the second inequality follows from triangle inequality, and the third from the union bound together with the fact that Y_{M^*} and Y_M are identically distributed. The last equality holds because the absolute value of a Laplace distribution follows an exponential distribution. We conclude that our solution is within an error α with probability at least $1 - 2\exp\left(-\frac{\epsilon\alpha}{2}\right)$, or equivalently, with probability at least $1-\beta$ our error is within $\frac{2}{\epsilon}\log\left(\frac{2}{\beta}\right)$.

By the same argument, the expected error will be at most

$$\mathbb{E}\left(|Y_{M^*}-Y_M|\right) \leq \mathbb{E}\left(|Y_{M^*}|+|Y_M|\right) = 2\mathbb{E}\left(|Y_M|\right) = \frac{2}{\epsilon}.$$

The last equality follows again from the exponential distribution of $|Y_M|$.

4 Conclusion and Open problems

This work shows that it is certainly feasible to design recommendation systems for online dating which satisfy differential privacy, and opens many directions for future research. First, the theoretical results of this paper should be tested on real data to verify their practicality. Secondly, finding a more efficient solution to the differentially private maximum matching problem is an interesting open question. Potentially strategies from approximation algorithms could be interesting, since they trade accuracy for efficiency and, in a differentially private setting, we are not looking for a perfectly accurate solution. Lastly, all results presented in this work model the problem to be static: we have a static data set and give a set of recommendation once. In practice, both users and edges will appear on or leave the platform, and we will give recommendations to users over a longer period of time. Specifically, this will make the assumption that the probabilities are independent for stochastic matching invalid. To capture these properties it would be necessary to consider dynamic models.

Acknowledgements. I want to thank Inge Li Gørtz, Philip Bille and Sune Lehmann for helpful suggestions and discussions.

References

1. B. Anandan and C. Clifton. Secure minimum weighted bipartite matching. In *Proc. DSC 2017*, pages 60–67, 2017.
2. X. Cai, M. Bain, A. Krzywicki, W. Wobcke, Y. S. Kim, P. Compton, and A. Mahidadia. Collaborative filtering for people to people recommendation in social networks. In *Proc. 23rd AI*, pages 476–485, 2010.
3. J. A. Calandrino, A. Kilzer, A. Narayanan, E. W. Felten, and V. Shmatikov. " you might also like:" privacy risks of collaborative filtering. In *Proc. 32nd IEEE Symposium on Security & Privacy*, pages 231–246, 2011.
4. N. Chen, N. Immorlica, A. R. Karlin, M. Mahdian, and A. Rudra. Approximating matches made in heaven. In *Proc. 36th ICALP*, pages 266–278, 2009.
5. C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *Proc. 24th EUROCRYPT*, pages 486–503, 2006.
6. C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. TCC 2006*, pages 265–284, 2006.

7. C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, 2014.
8. A. Friedman, S. Berkovsky, and M. A. Kaafar. A differential privacy framework for matrix factorization recommender systems. *User Model User-adapt Interact*, 26(5):425–458, 2016.
9. A. Friedman, B. P. Knijnenburg, K. Vanhecke, L. Martens, and S. Berkovsky. *Privacy Aspects of Recommender Systems*, pages 649–688. Springer US, Boston, MA, 2015.
10. R. Guerraoui, A.-M. Kermarrec, R. Patra, and M. Taziki. D 2 p: distance-based differential privacy in recommenders. *Proceedings of the VLDB Endowment*, 8(8):862–873, 2015.
11. M. Hay, C. Li, G. Miklau, and D. Jensen. Accurate estimation of the degree distribution of private networks. In *Proc. ICDM 2009*, pages 169–178, 2009.
12. K. He and X. Mu. Differentially private and incentive compatible recommendation system for the adoption of network goods. In *Proc. 15th ACM EC*, pages 949–966, 2014.
13. J. Hsu, Z. Huang, A. Roth, T. Roughgarden, and Z. S. Wu. Private matchings and allocations. *SIAM J. Comput*, 45(6):1953–1984, 2016.
14. Z. Ji, Z. C. Lipton, and C. Elkan. Differential privacy and machine learning: a survey and review. *arXiv preprint arXiv:1412.7584*, 2014.
15. S. Kannan, J. Morgenstern, R. Rogers, and A. Roth. Private pareto optimal exchange. *ACM Trans. Econ. Comput.*, 6(3-4):12, 2018.
16. X. Liu, A. Liu, X. Zhang, Z. Li, G. Liu, L. Zhao, and X. Zhou. When differential privacy meets randomized perturbation: A hybrid approach for privacy-preserving recommender system. In *Proc. 22nd DASFAA*, pages 576–591, 2017.
17. A. Machanavajjhala, A. Korolova, and A. D. Sarma. Personalized social recommendations: accurate or private. *Proc. of the VLDB Endowment*, 4(7):440–450, 2011.
18. F. McSherry and I. Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proc. 15th ACM SIGKDD*, pages 627–636, 2009.
19. F. McSherry and K. Talwar. Mechanism design via differential privacy. In *Proc. 48th FOCS*, volume 7, pages 94–103, 2007.
20. L. Pizzato, T. Rej, J. Akehurst, I. Koprinska, K. Yacef, and J. Kay. Recommending people to people: the nature of reciprocal recommenders with a case study in online dating. *User Model User-adapt Interact*, 23(5):447–488, 2013.
21. L. A. Pizzato, T. Rej, K. Yacef, I. Koprinska, and J. Kay. Finding someone you will like and who won't reject you. In *Proc. 19th UMAP*, pages 269–280, 2011.
22. L. A. Pizzato and C. Silvestrini. Stochastic matching and collaborative filtering to recommend people to people. In *Proc. 5th RecSys*, pages 341–344, 2011.
23. H. Shin, S. Kim, J. Shin, and X. Xiao. Privacy enhanced matrix factorization for recommendation with local differential privacy. *IEEE Trans. Knowl. Data Eng.*, 30(9):1770–1782, 2018.
24. T. A. Steiner. Differential privacy in graphs. Master's thesis, Technical University of Denmark, 2019.
25. T. Zhu, G. Li, W. Zhou, and S. Y. Philip. Differentially private data publishing and analysis: A survey. *IEEE Trans. Knowl. Data Eng.*, 29(8):1619–1638, 2017.
26. T. Zhu, Y. Ren, W. Zhou, J. Rong, and P. Xiong. An effective privacy preserving algorithm for neighborhood-based collaborative filtering. *Future Gener. Comput. Syst.*, 36:142–155, 2014.