Serenade: An Approach for Differentially Private Greedy Redescription Mining

Maiju Karjalainen, Esther Galbrun, and Pauli Miettinen[0000−0003−2271−316X]

University of Eastern Finland `firstname.lastname@uef.fi`

Abstract. We present an approach for mining differentially private redescriptions. Applying differential privacy to pattern mining approaches such as redescription mining is less straightforward than with numerical methods, in part because local patterns are more susceptible to the kind of noise inherent to differential privacy, and in part because the typically combinatorial algorithms require many operations and become computationally unaffordable when overloaded with privacy preserving procedures. Our solution, and the two algorithms we provide, address both the computational bottlenecks as well as the issues with noise. They might also provide inspiration for other differentially private pattern mining algorithms. Extensive experiments on real-world data validate the usefulness of our algorithms.

1 Introduction

As awareness about the rights of data subjects rises, so does the demand for privacy-preserving data mining methods. This trend, combined with concerns about black-box models being used in many domains, has lead to a need for methods that couple interpretability and privacy. Here we present algorithms answering this need, bringing differential privacy to redescription mining.

Differential privacy [5] is the most promising approach for privacy preserving data analysis. It allows to provide guarantees on the level of privacy offered, even in the presence of side information. Differentially private results are also secure against post-processing: no amount of post-processing will reveal any further private information from the results of differentially-private algorithms.

It is widely recognized that pattern and rule-based approaches, such as redescription mining [13], are highly interpretable data analysis approaches. However, integrating differential privacy with such approaches is not simple, as they are based on local properties of the data, which are more prone to privacy leakage than global properties. Moreover, the combinatorial nature of these approaches makes them typically more susceptible to noise than continuous optimization methods.

We tackle these obstacles and present two algorithms based on existing approaches for differential privacy and redescription mining: a greedy algorithm for redescription mining, and the exponential and Laplace mechanisms for differential privacy. Their successful combination requires developing novel methods. In

particular, we present a general technique to lower the sensitivity of the quality measure to magnitudes well below 1, allowing much more accurate results to be drawn from the exponential mechanism without compromising the privacy. In addition, we use weighted reservoir sampling for memoization, yielding an efficient implementation of the exponential mechanism.

2 Background and Related Work

Redescription mining. Redescription mining is a data analysis task that aims at finding two different ways to describe almost the same observations. In our setting, the data set $\mathcal{D}$ consists of a pair of tables, respectively denoted $\mathbf{D}_L$ and $\mathbf{D}_R$. Both tables are over the same entities (rows) but disjoint collections of attributes (columns).

Such a pair of data tables, along with user-defined parameters, constitute the input of redescription mining. A redescription consists of a pair of *queries* denoted by q_L and q_R, one over either data table. A query consists of literals over Boolean, categorical or numerical attributes, combined with logical conjunction and disjunction operators, possibly involving negations. For instance, the query

$$q = (\neg\,\mathrm{Hypertension} \wedge [\mathrm{State} = \mathrm{CA}]) \vee [\mathrm{BirthY} \leq 1950]$$

describes entities, here individuals, who either do not suffer from hypertension and reside in California, or are born before 1950. The collection of entities described by the queries is called the *support* of the query, denoted $\mathrm{supp}(q)$.

The main quality of a redescription is that the queries have similar supports. Referred to as its accuracy, it is measured using the Jaccard coefficient:

$$J(q_L, q_R) = |\mathrm{supp}(q_L) \cap \mathrm{supp}(q_R)| \,/\, |\mathrm{supp}(q_L) \cup \mathrm{supp}(q_R)| \ . \tag{1}$$

In addition to having a high accuracy, to be of interest a redescription should cover enough entities. This can be ensured by setting a minimum support threshold.

Since the introduction of redescription mining [26], various algorithms have been proposed for the task, including algorithms based on decision trees [26, 32, 24], on mining and combining frequent itemsets [14], or considering regression and correlation models [27]. We build on the ReReMi algorithm [10], that iteratively grows the queries in a greedy fashion. Further details can be found in [13].

Differential privacy. Differential privacy ensures that it cannot be inferred, within a fixed probability, whether or not a particular individual is included in a data set, regardless of what other information the adversary might have. This is achieved by withholding direct access to the data and only returning randomized outputs.

Formally, if $\mathcal{D}$ and $\mathcal{D}'$ are two data sets that differ in exactly one row, A is a randomized algorithm operating on these tables, S is any set of potential results A might return, and $\varepsilon \in [0, 1]$ is the privacy parameter (or budget), then A is ε-*differentially private* (later also simply ε-private) if

$$\Pr[A(\mathcal{D}) \in S] \leq e^{\varepsilon} \Pr[A(\mathcal{D}') \in S] \ , \tag{2}$$

Algorithm 1 SERENADECS, client algorithm with client-side decisions

---

Input: Privacy budget $\varepsilon_{\text{tot}} = \varepsilon_{\text{init}} + \varepsilon_{\text{ext}} + \varepsilon_{\text{qual}}$, number of initial pairs I, number of
extensions L, minimum support z_o, minimum accuracy $J_{\min}$, steepness s
Output: A set of redescriptions together with their accuracy $\mathcal{R} = \{(q_L, q_R, J)\}$
 1: **function** SERENADECS($\varepsilon_{\text{init}}, \varepsilon_{\text{ext}}, \varepsilon_{\text{qual}}, I, L, z_o, J_{\min}, s$)
 2:     $\mathcal{R} \leftarrow \emptyset$
 3:     $\mathcal{Q} \leftarrow$ INTIALPAIRS($\varepsilon_{\text{init}}/2, I, z_o, s$)
 4:     **for** $r_0 \in \mathcal{Q}$ **do**
 5:         $\ell \leftarrow 0$
 6:         $J_0 \leftarrow$ COMPUTEQUALITY($r_\ell, \varepsilon_{\text{qual}}/(L|\mathcal{Q}|)$)
 7:         **while** $\ell < L$ and $J_\ell > J_{\min}$ **do**
 8:             $\ell \leftarrow \ell + 1$
 9:             $r_\ell \leftarrow$ EXTENDONE($\{r_{\ell-1}\}, \varepsilon_{\text{ext}}/(2L|\mathcal{Q}|), z_o, s$)
10:             $J_\ell \leftarrow$ COMPUTEQUALITY($r_\ell, \varepsilon_{\text{qual}}/(L|\mathcal{Q}|)$)
11:         Add to $\mathcal{R}$ the last r_ℓ with $J_\ell > J_{\min}$, if any
12:     **return** $\mathcal{R}$

---

where the probability is over the randomness of A (see [6]).

The core mechanism for ensuring differential privacy with numerical outputs is the *Laplace mechanism*, that perturbs numerical outputs with *Laplace*$(0, \Delta A/\varepsilon)$-distributed noise, where ΔA is the *sensitivity* of A, i.e. the maximum change in the output of A given two inputs that differ in exactly one row. For other than numerical outputs, the *exponential mechanism* [23] is used. The space of all possible outputs of A must be endowed with a data-specific *quality function* $f(A(\mathcal{D}))$, and the answer is sampled from this space with probability $\propto \exp(\varepsilon \cdot f(A(\mathcal{D})))$. This ensures $2\varepsilon\Delta f$-privacy [23]. Further developments in differential privacy include concentrated differential privacy [7], which trades better behaviour under composition to weaker definition of privacy. Our method is trivially $(\varepsilon \cdot (e^\varepsilon - 1)/2, \varepsilon)$-concentrated differentially private [7], but further analysis of concentrated differential privacy is beyond the scope of this paper.

After the initial formulation [5] and the introduction of the exponential mechanism [23], differential privacy has seen significant interest in academia, and recent years have brought uptake by the industry as well (e.g. [4]). Real-world use of differential privacy has indicated that the typical assumption that $\varepsilon < 1$ is often unachievable (e.g. the US Census Bureau used $\varepsilon = 19.61$ for releasing the 2020 US Census results [29]). Differentially private algorithms have been proposed for mining frequent itemsets [31, 20], association rules [28, 22], as well as subgraphs [30] or sequences [3].

3   The Algorithms

In this section we present two algorithms for differentially private redescription mining: SERENADECS and SERENADEES. Both algorithms are based on the greedy redescription mining approach of REREMI and assume that one side of

Algorithm 2 SERENADEES, client algorithm with private extensions

---

Input: Privacy budget $\varepsilon_{\mathrm{tot}} = \varepsilon_{\mathrm{init}} + \varepsilon_{\mathrm{ext}} + \varepsilon_{\mathrm{qual}}$, number of initial pairs I, number of
 extensions K, minimum support z_o, steepness s
Output: A set of redescriptions together with their accuracy $\mathcal{R} = \{(q_L, q_R, J)\}$

 1: **function** SERENADEES$(\varepsilon_{\mathrm{init}}, \varepsilon_{\mathrm{ext}}, \varepsilon_{\mathrm{qual}}, I, K, z_o, s)$
 2: $\mathcal{Q} \leftarrow$ INTIALPAIRS$(\varepsilon_{\mathrm{init}}/2, I, z_o, s)$
 3: **for** $k = 1, \ldots, K$ **do**
 4:   $r \leftarrow$ EXTENDONE$(\mathcal{Q}, \varepsilon_{\mathrm{ext}}/2K, z_o, s)$
 5:   $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{r\}$
 6:   Remove extended redescription from $\mathcal{Q}$
 7: $\mathcal{R} \leftarrow \emptyset$
 8: **for all** $r \in \mathcal{Q}$ **do**
 9:   $\mathcal{R} \leftarrow \mathcal{R} \cup \{(r, \text{COMPUTEQUALITY}(r, \varepsilon_{\mathrm{qual}}/|\mathcal{Q}|))\}$
10: **return** $\mathcal{R}$

---

the data has only Boolean and categorical attributes, a common assumption in redescription mining (see, e.g. [10]).

Both algorithms consist of a client with no direct access to the data and a differentially private engine, with full access to the data, that the client queries. The two algorithms differ mainly on their client-side implementations, i.e. how they use the differentially private engine's methods. With SERENADECS, more of the decisions are made on the client side (CS), resulting in the ability to prune low-quality results at the cost of an increased budget expenditure. SERENADEES provides a thriftier alternative by letting the differentially private engine make the decisions (engine side, ES).

We will next explain how the clients work before explaining how the differentially private part has been implemented.

3.1 Client side

The pseudocode for the algorithms is given in Algorithms 1 and 2. Both algorithms mine redescriptions through three main steps. The INTIALPAIRS method (Algorithm 4) returns redescriptions with a single literal in both queries. These pairs are then iteratively extended into longer redescriptions. The EXTENDONE method (Algorithm 3) takes a set of redescriptions as input and returns a new redescription, obtained by extending one of the input redescriptions by one literal. The COMPUTEQUALITY method computes the noisy Jaccard coefficient and support size of a redescription. The two algorithms differ on when the quality of redescriptions is computed and how to select the next redescription to extend.

SERENADECS closely follows the approach of REREMI: it considers one initial pair at a time (line 4 in Algorithm 1) for at most a fixed number of extensions (line 7). In each iteration, SERENADECS calls EXTENDONE with the current candidate as input (line 9), queries COMPUTEQUALITY to get the noisy Jaccard of the extension (line 10), and decides whether to continue with the extension process. This ensures we only attempt to extend redescriptions that are good

Algorithm 3 EXTENDONE, extending a redescription
___
Input: Data $\mathcal{D}$, collection of redescriptions $\mathcal{Q}$, privacy budget ε, minimum support z_o,
    steepness s
Output: Extended redescription (q'_L, q'_R)
Private information: Data $\mathcal{D} = (\mathbf{D}_L, \mathbf{D}_R)$, sensitivity Δf given $\mathcal{D}$, z_o, and s
Static variables: Associative array M, priority queue P
 1: **function** EXTENDONE($\mathcal{Q}, \varepsilon, z_o, s$)
 2:    **for all** $(q_L, q_R) \in \mathcal{Q} \setminus M$ **do**
 3:        **for all** one-literal extensions (q'_L, q'_R) **do**
 4:            select the extensions with the highest key $key(f(q_L, q_R, z_o, s), \varepsilon/\Delta f)$
 5:        $M[(q_L, q_R)] \leftarrow (q'_L, q'_R)$ with age 1
 6:        $P.\texttt{push}(k, (q_L, q_R))$
 7:    $k, (q_L, q_R) \leftarrow P.\texttt{pop}()$
 8:    $(q'_L, q'_R) \leftarrow M[(q_L, q_R)]$; delete $M[(q_L, q_R)]$
 9:    increment the age of every $(q_L, q_R) \in M$ and remove expired from M and P
10:    **return** (q'_L, q'_R)
___

enough, but it has two drawbacks: *i)* it needs to call COMPUTEQUALITY for every extension, using plenty of budget; *ii)* it might attempt to extend a redescription which has no good extensions, effectively wasting the associated budget. The number of redescriptions returned by SERENADECS is not predetermined, though it is never larger than the number of initial pairs I.

SERENADEES addresses these drawbacks of SERENADECS. At each of a fixed number of extensions, it sends *all* current candidate redescriptions that can be extended to EXTENDONE and receives one extension back (line 4 in Algorithm 2). The decision of which redescription to extend is made in the differentially private engine. Furthermore, the Jaccard of the candidates is only computed for the redescriptions that are returned (line 9). This approach saves on the privacy budget compared to SERENADECS. It also allows some redescriptions to be extended multiple times while not wasting budget on redescriptions that cannot be extended. On the flip side, EXTENDONE might return a bad redescription, and this will only become apparent when it is too late to replace the candidate. SERENADEES always returns as many redescriptions as there were initial pairs.

3.2 Differentially Private Engine

Next, we present the three methods of the differentially private engine. We start with a general overview and description of the main ingredients of our algorithms.

Reservoir sampling. Greedy extension is a standard strategy for building redescriptions [13]. Extending redescription (q_L, q_R) consists in appending a literal to either of the queries using a disjunction or conjunction. The extension with the highest accuracy that satisfies constraints (e.g. minimum support threshold) is reported. For numerical attributes, all relevant intervals are evaluated in turn. A pruning trick [10] allows to identify relevant intervals, i.e. those that impact the accuracy, and avoid testing those that do not (see Section 3.3).

Algorithm 4 INTIALPAIRS, generating initial candidates

Input: Privacy budget ε, number of initial pairs I, minimum support threshold z_o, steepness s

Output: A set of initial redescriptions $\mathcal{R} = \{(q_L, q_R)\}$

Private information: Data $\mathcal{D} = (\mathbf{D}_L, \mathbf{D}_R)$, sensitivity Δf given $\mathcal{D}$, z_o and s

 1: **function** INTIALPAIRS(ε, I, z_o, s)
 2:     $\mathcal{R} \leftarrow \emptyset$
 3:     Set S_i as size-1 reservoir sampler for $i = 1, \ldots, I$
 4:     **for all** literals x over the non-numerical attributes of $\mathbf{D}_L$, numerical attributes y in $\mathbf{D}_R$, and meaningful intervals $[\lambda, \rho]$ of y **do**
 5:         Add initial pair $(x, [\lambda \leq y \leq \rho])$ to all reservoir samplers S_i where $key\big(f(x, [\lambda \leq y \leq \rho], z_o, s), \varepsilon/(I\Delta f)\big)$ is greater than the current key
 6:     **for all** reservoirs S_i **do**
 7:         Add redescription from S_i to $\mathcal{R}$
 8:     **return** $\mathcal{R}$

A differentially private variant of such an extension strategy is best implemented with the exponential mechanism. Instead of reporting the best extension, the algorithm will randomly pick one of the candidates with probabilities proportional to their accuracies. The set of candidates will also include the current, unextended, redescription associated to a sampling probability reflecting its quality. That way, the structure of the queries will not reveal anything sensitive about the data (see Section 3.3 about the case of continuous-valued attributes).

A naïve approach would be to store all candidates, but this would use too much memory. Instead, we use *weighted reservoir sampling* [8, 19]. The idea is to keep a reservoir of size k and evaluate candidate extensions in turn. For each one, we sample a value $u \sim Unif_{[0,1]}$. If the probability of the candidate being selected by the exponential mechanism is α, we compute its key as $u^{1/\alpha}$ and store the candidate in the reservoir if this key is larger than the current smallest key in the reservoir (see [8, 19] for proofs of correctness). That is, for each candidate (q_L, q_R) we compute

$$key(f(q_L, q_R, \Theta), \varepsilon) = \log(Unif_{[0,1]}) \cdot \exp\big(\varepsilon \cdot f(q_L, q_R, \Theta)\big) , \qquad (3)$$

where $f(q_L, q_R, \Theta)$ is a function measuring the quality of (q_L, q_R) given parameters in Θ, and retain the candidates with highest keys.

Extending a redescription. When called from SERENADEES, EXTENDONE (Algorithm 3) receives a collection of redescriptions $\mathcal{Q}$. In each call, the method samples from the distribution of all extensions of all redescriptions in $\mathcal{Q}$, and one selected candidate replaces the extended redescription. This proceeds for a fixed number of iterations, progressively extending and replacing redescriptions in $\mathcal{Q}$. Between two successive calls to EXTENDONE, only one redescription changes.

Our sampling approach also allows us to first draw a large sample and then take a smaller sample from it, by simply retaining the entries associated with

the largest keys. As the collection $\mathcal{Q}$ can be large, we use this strategy to save computational time.

Specifically, EXTENDONE stores for each redescription encountered so far the sampled extension for that redescription, together with its key. When a new redescription is added to $\mathcal{Q}$, EXTENDONE samples an extension for it and stores it. The redescription whose sampled extension has the largest key is selected and the corresponding extended redescription is returned.

This memoization can lead to suboptimal choices if an extension of a redescription gets a very low key value even though it is relatively good, and vice versa. To counteract that, EXTENDONE prunes stored extensions that are too old, i.e. extensions that have not been selected in the past extension rounds. The maximum age of an extension is a user-supplied parameter. When called from SERENADECS, EXTENDONE operates on a single redescription and returns its candidate extension having the highest key, without any memoization.

Computing initial pairs and redescription quality. INTIALPAIRS (Algorithm 4) builds initial pairs by enumerating all pairs of attributes (x, y), where x is an attribute from $\mathbf{D}_L$ and y is an attribute from $\mathbf{D}_R$. The selection of the initial pairs from all pairs is based on a reservoir sampling approach. Assuming w.l.o.g. that the non-numerical attributes are in $\mathbf{D}_L$, we consider as initial query q_L every literal consisting of a Boolean attribute or an attribute–category pair. For q_R, we consider every literal (including all relevant intervals) from $\mathbf{D}_R$ to match q_L, in the same way as when extending an existing redescription. The sampling is done over the queries q_R. In order to generate I initial pairs, we populate I different size-1 reservoirs, corresponding to I independent samples with replacement from the exponential mechanism, ensuring privacy.

COMPUTEQUALITY computes the size of the intersection and the size of the union of the input queries' supports and returns them after applying Laplace noise. The noisy Jaccard of the redescription is computed as the noisy intersection size divided by the noisy union size.

3.3 Handling Continuous Attributes

Continuous attributes require extra care in both computational efficiency and privacy. When appearing in a query, a continuous attribute $y \in \mathbb{R}$ will be bounded within an interval, as $[\lambda \leq y \leq \rho]$ (where either λ or ρ can be infinite). Finding these intervals efficiently is the first problem. We follow the cut-point-based approach of REREMI [10]. This approach is based on the fact that only some values of y will have an effect on the support of the extended query when used as threshold. If an entity is not in $\text{supp}(q_R)$, whether or not it belongs to $\text{supp}([\lambda \leq y \leq \rho])$ is indifferent w.r.t. the support of the extended query. Hence we only need to consider as thresholds those values that determine whether entities in $\text{supp}(q_R)$ are included in $\text{supp}([\lambda \leq y \leq \rho])$. These values, along with $-\infty$ and ∞, are called *cut-points*.

This approach extends to disjunctions and negations. It also helps us with privacy. Returning the exact value of cut-points can leak information about the

data, as it reveals that y takes this exact value at least once.[1] Let us assume that the lower cut-points and upper cut-points are $-\infty = \lambda_1 < \lambda_2 < \cdots < \lambda_l$ and $\rho_1 < \rho_2 < \cdots < \rho_p = \infty$, respectively, and that we have chosen to use interval $[\lambda_i \leq y \leq \rho_j]$. Instead of reporting λ_i, we sample the value λ_i' we report as lower threshold from $Unif_{(\lambda_{i-1}, \lambda_i]}$ and instead of ρ_j, we sample a value ρ_j' from $Unif_{[\rho_j, \rho_{j+1})}$. Note that if $i = 1$ or $j = p$, we actually have a half-interval, which we write $[y \leq \lambda_j']$ or $[\rho_j' \leq y]$, and no sampling is necessary for that threshold. This sampling ensures that the accuracy of the redescription remains the same while no information is leaked about the exact values in the data.

3.4 Lower-Sensitivity Quality Function

A key element of the algorithm is the distribution of the budget between the steps. Both INTIALPAIRS and EXTENDONE use the exponential mechanism, which provides $2\varepsilon \Delta f$-privacy for quality f of sensitivity Δf. For our application, the standard quality function is the Jaccard coefficient, which has sensitivity 1, since it takes values between 0 and 1. However, for interesting redescriptions, the support size is well above 0 and, as a result, adding or removing a single row will not change the Jaccard coefficient much. This motivates us to modify the quality function to take into account the support size. Specifically, we multiply the Jaccard coefficient with a logistic function centered at the user-defined minimum support threshold z_o. The quality function f becomes

$$f(q_L, q_R, z_o, s) = J(q_L, q_R) / \left(1 + \exp\left(-s \cdot (|\operatorname{supp}(q_L) \cap \operatorname{supp}(q_R)| - z_o)\right)\right), \quad (4)$$

where s is a parameter adjusting the steepness of the logistic curve.

For large values of s the sensitivity of f will still be 1. For smaller values of s, however, it is possible to significantly reduce the sensitivity of f. Given z_o, s, and data set size n, the sensitivity of f can be calculated in $O(n^2)$ time by considering all possible values for numerator and denominator in (1), plugging them into (4), and identifying the largest change that can result from the addition of a single row. Furthermore, the steepness can be optimized with a simple line search over different sensitivity parameters. With the data sets and parameters used in our experiments, the sensitivity was always less than 0.001, leaving more budget to be spent on initial pairs and extensions.

Notice that knowing the sensitivity releases information about the data as it depends on the data set size. This leaves a few options. The first is that the differentially private engine computes the optimal steepness and uses the corresponding sensitivity when distributing the budget. This approach is completely private, but means that the user has no knowledge of the actual noise levels used. The other extreme is to use a weaker form of differential privacy, where the user knows the size n of the data set. The user can then find the optimal s minimizing the sensitivity of (4). As an intermediate option, the user might query for n, use the noisy estimate to compute near-optimal steepness, and give that

---

[1] We assume that the fact that categorical attributes take each category at least once is not private information.

as a parameter to the algorithm. This way, at the cost of some privacy budget, the user retains some knowledge of the quality function and knows its steepness, but no privacy is lost.

3.5 Other Parameters

The two key parameters of the algorithms are the number of initial pairs, I, and the number of extensions, K or L. The higher I is set, the more redescriptions will be returned, but each of them will receive a smaller portion of the privacy budget, both when generating the initial pairs and when computing the privatized supports and accuracies. On the other hand, the number of extensions (K or L) dictates how complex the redescriptions can become. SERENADECS will return no more than I redescriptions, and none of them will have more than L extensions (i.e. they have at most $L + 2$ literals in total). SERENADEES will return exactly I redescriptions, but some of them might have just two literals, or some of them might have $K + 2$ literals. Hence, if SERENADECS is set to perform L extensions, a comparable setting for SERENADEES is $K = I \cdot L$ extensions.

3.6 Privacy

The privacy of the proposed approach is mainly based on the exponential mechanism. For this, we need to sample from every possible extension with the correct probability. Next we show that this is the case. We then continue by analysing the use of the budget, and conclude that the methods are indeed private.

Lemma 1. *Given a redescription $r_{\ell-1} = (q_L, q_R)$, the probability that* EXTEND-ONE *in line 9 of Algorithm 1 returns an extension $r_\ell = (q'_L, q'_R)$, where either i) q'_L is an arbitrary single-literal extension of q_L and $q_R = q'_R$, ii) q'_R is an arbitrary single-literal extension of q_R and $q_L = q'_L$, or iii) $q_L = q'_L$ and $q_R = q'_R$, is $\propto \exp(\varepsilon \cdot f(q'_L, q'_R, z_o, s))$.*

Proof. We can assume w.l.o.g. that $q_L = q'_L$. Assume first that $\mathbf{D}_R$ does not contain continuous attributes. To select the extension of q_R, EXTENDONE can be considered to go through every extension, computing their quality according to f, and placing the extension into a stream together with a weight parameter that is set to $\exp(\varepsilon \cdot f)$. Finally, the algorithm will also place the unextended q_R into the stream with weight corresponding to the quality of (q_L, q_R). The weighted reservoir sampler will select one item from this stream, so that the probability of an item being selected is proportional to its weight [19, 8], thus sampling the extension with correct probability.

For continuous attributes, the algorithm further perturbs the boundaries as explained in Section 3.3. As this does not affect the quality, all extensions with the same quality have equal probability to be selected, concluding the proof. $\square$

Lemma 2. *The extended redescription $r = (q'_L, q'_R)$ returned by* EXTENDONE *when called in line 4 of Algorithm 2 is selected with probability that is proportional to $\exp(\varepsilon \cdot f(q'_L, q'_R, z_o, s))$ from all possible at-most-single literal extension to all redescriptions in $\mathcal{Q}$.*

Lemma 3. *Given a pair of single-literal queries (q_L, q_R) over data sets $\mathbf{D}_L$ and $\mathbf{D}_R$, the probability that INTIALPAIRS returns (q_L, q_R) is $\propto I \cdot \exp(\varepsilon \cdot f(q_L, q_R, z_o, s))$.*

Lemma 4. *The total budget used by SERENADECS and SERENADEES never exceeds ε_{tot}.*

The proof for Lemma 3 follows the same lines as for Lemma 1, except that the sampling is repeated I times, and is omitted. The proofs for Lemma 2 and Lemma 4 are postponed to the full version of the paper. [2]

Theorem 1. *SERENADECS and SERENADEES are ε_{tot}-differentially private.*

Proof. Per Lemma 3, both algorithms generate initial pairs privately. Each initial pair is an independent query to the data, spending altogether budget $\varepsilon_{\mathrm{init}}$. Every extension in SERENADECS is an independent, differentially private query (Lemma 1). Due to the composability of differential privacy [6], they do not reveal more information than permitted by their allocated budget. The same holds for extensions in SERENADEES. Finally, quality calculations in COMPUTEQUALITY use the standard Laplace mechanism and are private, in line with their allocated budget. The total budgets do not exceed the allocated $\varepsilon_{\mathrm{tot}}$ (Lemma 4). □

4 Experimental Evaluation

To test our proposed methods, we conducted both quantitative and qualitative experiments. The purpose of the quantitative experiments is to investigate the behavior of the algorithms with different parameters and carry out an ablation study. The qualitative experiments are meant to assess the usefulness of our approach for real-world data analysis. For this reason, we did not do any parameter tuning as that would consume some of the privacy budget.

All algorithms are implemented in Python and the experiments are run with Python 3.6.8 on a machine with 2 AMD EPYC 7702 processors with 64 cores each and 1 TB of main memory.

4.1 Data and Other Methods

The `Mammals` data set contains information about which mammal species inhabit which areas of Europe on one side [25], and climate information on the other side [16]. This data set has often been used in redescription mining literature (e.g. [10, 11, 32, 18, 24, 12]). The data contains 194 species and 48 climate attributes as its columns, with 2575 localities as its rows.

The `MIMIC` data set [17][3] is an example of health data where patient privacy is critical. It contains de-identified health information about hospital patients.

---

[2] The source code and the appendix containing all details that are postponed to the full version of the paper are presented in `https://github.com/maijuka/Serenade`

[3] Data set available at `https://physionet.org/content/mimiciii/1.4/`.

Our dataset contains two views: the diagnoses view and the lab events view. The aim when mining this data set is to find out what kind of lab events are associated with various diagnoses of the examined patients. The data set contains 107 Boolean attributes in the diagnoses view (1 denotes a positive diagnosis) and 104 ternary attributes in the lab events view, with 46 065 patients as its rows.

The `VAA` data set [15] contains the background information and answers to questions regarding political opinions of candidates to the Finnish parliamentary elections of 2011. The data was collected by an online voting advice application (VAA). We removed candidates who did not provide answers to all questions, leaving 1656 candidates, 9 background attributes and 107 opinion attributes.

The Open Sex-Role Inventory [21] (`SR` data set) is used to study gender roles by measuring masculinity and femininity. The Right-Wing Authoritarianism Scale [1] (`RightW` data set) was developed to measure the psychology of the followers of facist regimes.[4] Columns recording to the time spent answering the questions were removed from the `RightW` and the `SR` data sets. These data sets have the same variables on both sides; INTIALPAIRS and EXTENDONE are implemented to ensure the same attribute cannot be used in both queries of the same redescription.

We also conducted experiments with other datasets. The results were not significant and are postponed to the full version of the paper.

The main studied algorithms are SERENADECS and SERENADEES. In addition, for ablation studies, we used a variation of SERENADEES called NOSENS. It works like SERENADEES, but uses the standard Jaccard (1) instead of the lower-sensitivity version (4) for computing keys (3). We also did a comparison to REREMI, the baseline non-private redescription mining algorithm [10].

For all experiments, we set $J_{\min} = 0.3$ for SERENADECS and z_o to 10% of data set size n.

4.2 Quantitative Experiments

In this empirical evaluation, we investigate the effects of parameters and carry out an ablation study using NOSENS.

Privacy budget. A typical value for the privacy budget is $\varepsilon_{\text{tot}} = 1$, though much higher total budgets are sometimes used (e.g. [2, 9, 29]). Hence we first studied the effect of varying the total budget ε_{tot} along with different ways to distribute it between the different tasks. Total budgets of 0.5, 1, 10, and 100 were used. Of these, $\varepsilon_{\text{tot}} = 0.5$ and $\varepsilon_{\text{tot}} = 1$ are typical values found in the literature, $\varepsilon_{\text{tot}} = 10$ is closer to values used in practice [29], and $\varepsilon_{\text{tot}} = 100$ can be seen as a baseline of how the algorithms perform when privacy requirements are lifted.

Another parameter to consider is the distribution of the total budget between $\varepsilon_{\text{init}}$, ε_{ext}, and $\varepsilon_{\text{qual}}$. We used five different ways of distributing ε_{tot} between $\varepsilon_{\text{init}}$, ε_{ext}, and $\varepsilon_{\text{qual}}$, respectively in the following proportions: $d_1 = (0.33, 0.33, 0.33)$, $d_2 = (0.45, 0.45, 0.1)$, $d_3 = (0.6, 0.3, 0.1)$, $d_4 = (0.3, 0.6, 0.1)$, and $d_5 = (0.25, 0.25, 0.5)$. The maximum age of an extension was set to 40, $I = 20$, $L = 4$, and $K = 80$. Each experiment was repeated five times.

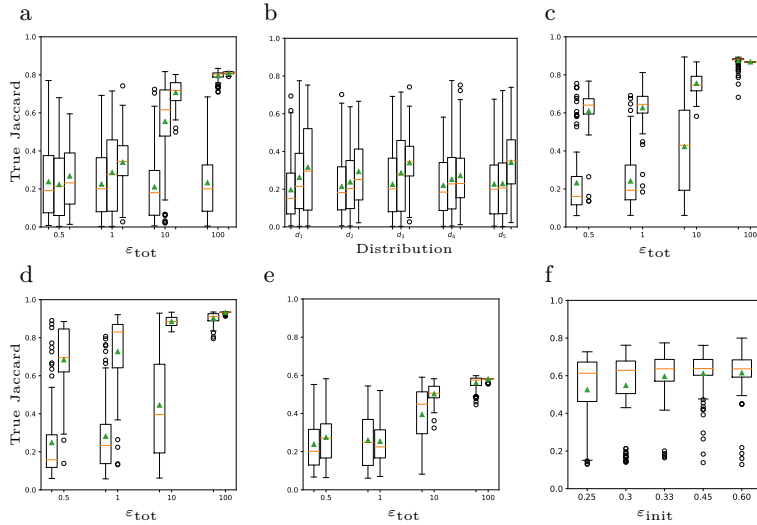[4] Data sets available at `https://openpsychometrics.org/_rawdata/`.

Fig. 1. Effects of the budget on the accuracy. y-axis is true Jaccard. x-axis: (a) Varying ε_{tot}, budget distribution d_3. (b) Varying budget distribution, $\varepsilon_{\text{tot}} = 1$. Boxes represent (left to right) NoSens, SerenadeES, and SerenadeCS. (c–e) same as in (a) with data sets (c) SR, (d) `RightW` and (e) `VAA`, but without NoSens. (f) Accuracy of the initial pairs, varying $\varepsilon_{\text{init}}$, with $\varepsilon_{\text{tot}} = 1$. The horizontal bar: median, triangle: mean.

Results for these experiments on the `Mammals` data set are presented in Fig. 1(a–b) where we see that both SerenadeCS and SerenadeES are capable of returning very good results when allocated sufficiently high budgets. On the other hand, NoSens struggles even when allocated very high budgets. Given its inefficient use of the budget, this is not surprising. Results for similar experiments on three other data sets are presented in Fig. 1(c–e) where we see that SerenadeCS returns clearly better results with almost all parameters for these datasets. We can also see that `VAA` (Fig 1(e)) yielded generally poor results regardless of budget and algorithm. This is explained by the fact that the non differentially private redescription mining algorithm ReReMi[5] also fails to mine good quality redescriptions from these data sets. These results are postponed to the full version of the paper. From Fig. 1(f) we see that we get slightly better initial pairs with a higher budget, which could be the reason why the distributions d_3 and d_2, having the highest budgets for initial pairs, perform slightly better than the other budget distributions.

SerenadeES typically has higher variance on the accuracy of the results, giving accuracies both higher and lower than SerenadeCS. This is in line with the expected behavior. Of the different budget distributions, d_3 seems to be consistently giving the best results, although the differences are not significant.
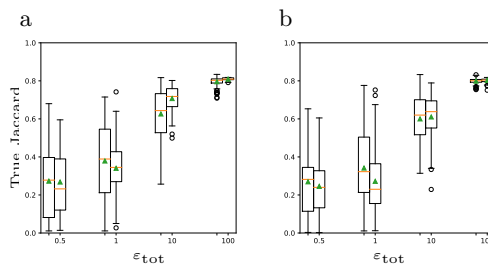
---

[5] Source code available at `https://pypi.org/project/python-siren/` [10]

Fig. 2. Effects of pruning results with low noisy Jaccard. (a) Varying ε_{tot}, budget distribution d_3. (b) Varying ε_{tot}, budget distribution d_1. Left box: SERENADEES, right box: SERENADECS. The horizontal bar: median, triangle: mean.

An analyst is typically not interested in results with very low reported accuracy (noisy Jaccard). Figure 2 repeats Fig. 1(a–b) for SERENADEES and SERENADECS after removing redescriptions with noisy Jaccard below 0.3. The average and median for SERENADEES improve, and especially the lower end of the tail rises, as expected. We can also see that this pruning leaves some redescriptions with *true* accuracy below 0.3, but does not actually discard many of those reported to be good. The discrepancy between the reported and true accuracy is an important feature: if the system reports a redescription as having a high accuracy, the user should be able to trust that it truly is so.

Number of initial pairs and reliability of noisy Jaccards. Next we investigate the effects of the number of initial pairs. For these tests, the maximum age was always set to be twice the number of initial pairs, and the number of extension rounds was four times the number of initial pairs. Budget distribution d_1 was used throughout. Recall that more initial pairs means the budget is divided between more redescriptions. This shows especially strongly in the differences between the true and reported accuracies (Figure 3).

Figure 3 also shows that SERENADECS returns fewer redescriptions than SERENADEES and that it occasionally reports overly optimistic Jaccard values even with $I = 20$. This is due to its mediocre use of the budget, leaving just slivers for relevant quality computations. Note that reported accuracies for SERENADECS are cut above $J_{\min} = 0.3$ through filtering. With 80 initial pairs, reported Jaccards become less correlated with true Jaccards for both methods, as budget per redescription dwindles.

Maximum age. The final quantitative study probes the effects of the maximum age memoization in EXTENDONE in SERENADEES. We investigated this by setting $K = 80$ and trying maximum ages 0 (computing extensions anew in every iteration), 5, 10, 20, 40, 60, and 81 (stored extensions never expire). We found that the memoization has a clear beneficial effect on the running time. Using maximum age 20 brings the running time down to about 25 min, from over 70 min without memoization (maximum age 0). Most redescriptions are selected for
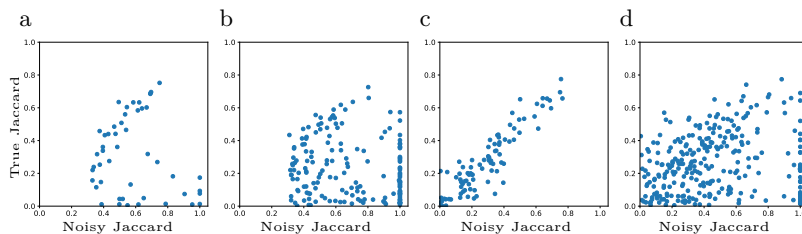
Fig. 3. Effects of the number of initial pairs. The y-axis represents the true Jaccard whereas the x-axis represents the noisy Jaccard. (a) SERENADECS and $I = 20$. (b) SERENADECS and $I = 80$. (c) SERENADEES and $I = 20$. (d) SERENADEES and $I = 80$.

extension within about 20 steps, so the running time does not improve noticeably beyond that. On the other hand, the maximum age did not seem to have a significant effect on the quality.

4.3   Qualitative Experiments

Next, we present a few results mined from different datasets, as anecdotal examples of results one can expect by our algorithms with these kinds of data sets. We did five runs of the algorithms for the `MIMIC`, `VAA`, `RightW`, `SR`, `SSC`, `Psycho`, and `Pref` data sets with total budget 1 and budget distribution d_3. The results were chosen over five runs for both algorithms, so the total budget per data set is 10. The results with `VAA`, `SSC`, `Psycho` and `Pref` datasets, and the raw redescriptions are postponed to the full version of the paper.

An example result (q_L, q_R) from `MIMIC` with SERENADECS is

$$(\text{Atrial Fibrillation}, \quad \text{Uric Acid} = 1 \vee \text{Creatine Kinase} = 0),$$

meaning that the set of patients who had atrial fibrillation (a type of arrhythmia, abnormal heart rhythm) corresponds to the set of patients whose uric acid (waste product in blood) test results were abnormal or whose creatine kinase (enzyme related to energy production) test results returned normal values. The reported accuracy of this redescription is 0.424 and its true accuracy is 0.227. Such results can be used to understand the procedures at this hospital, and as they are privatized, the results can also be communicated more widely without putting the privacy of patients in jeopardy.

A result with SERENADEES from the `RightW` data set tells that the set of participants who moderately or strongly agreed that "You have to admire those who challenged the law and the majority's view by protesting for women's abortion rights, for animal rights, or to abolish school prayer" corresponds to the set of participants who did not vote. The reported accuracy of this redescription is 0.428 and its true accuracy is 0.538.

A result from the `SR` data set with SERENADECS the set of participants who agreed or slightly agreed that they are the happiest when they are in their bed corresponds to the set of participants who agreed or slightly agreed that they

give people handmade gifts and are between 14 and 32 years of age. The reported accuracy of this redescription is 0.651 and its true accuracy is 0.725.

5 Conclusions

It is the nature of differential privacy and local pattern mining that the user can never be entirely certain whether a pattern reported by the algorithm actually represents a strong pattern in the data. But careful selection of the parameters allows to avoid most of the pitfalls, and when a privacy budget higher than 1 is available, the problem essentially vanishes. Even with a privacy budget of at most 1, the results can be very useful. Of the two approaches presented, SERENADECS gives results with higher (true and reported) Jaccard, but with lower correlation between the two, meaning that a good-looking result from SERENADECS might turn out not to be so good in reality. SERENADEES, on the other hand, returns more truthful Jaccards, thanks to its better use of budget, but its extension method might lead to generally somewhat inferior results.

Overall, the reported strong patterns are reliably present in the data. This means that a user who obtains interesting strong patterns via these differentially private approaches can seek further means to confirm them, for example by applying for a research permit for the data set or by conducting a separate confirmatory study. The techniques presented in this paper might also be used to implement other differentially private pattern mining algorithms in the future.

References

1. Altemeyer, B.: Right-wing authoritarianism. Univ. of Manitoba Press (1983)
2. Balu, R., Furon, T.: Differentially private matrix factorization using sketching techniques. In: IH&MMSec'16. pp. 57–62 (2016)
3. Bonomi, L., Xiong, L.: A two-phase algorithm for mining sequential patterns with differential privacy. In: CIKM'13. pp. 269–278 (2013)
4. Ding, B., Kulkarni, J., Yekhanin, S.: Collecting Telemetry Data Privately. In: NIPS'17. pp. 3571–3580 (2017)
5. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: TCC'06. pp. 265–284 (2006)
6. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
7. Dwork, C., Rothblum, G.N.: Concentrated differential privacy. arXiv preprint arXiv:1603.01887 (2016)
8. Efraimidis, P.S., Spirakis, P.G.: Weighted random sampling with reservoir. Inf. Proc. Lett. **97**, 181–185 (2006)
9. Fletcher, S., Islam, M.Z.: Decision tree classification with differential privacy: A survey. ACM Comput. Surv. **52**(4), 83:1–83:33 (2019)
10. Galbrun, E., Miettinen, P.: From black and white to full color: Extending redescription mining outside the Boolean world. Stat. Anal. Data Min. **5**(4), 284–303 (2012)
11. Galbrun, E., Miettinen, P.: Siren: An interactive tool for mining and visualizing geospatial redescriptions [demo]. In: KDD'12. pp. 1544–1547 (2012)

12. Galbrun, E., Miettinen, P.: Mining Redescriptions with Siren. ACM Trans. Knowl. Discov. Data **12**(1), 6 (2018)
13. Galbrun, E., Miettinen, P.: Redescription Mining. Springer (2018)
14. Gallo, A., Miettinen, P., Mannila, H.: Finding subgroups having several descriptions: Algorithms for redescription mining. In: SDM'08. pp. 334–345 (2008)
15. Helsingin Sanomat: Parliamentary elections 2011: Candidate responses to helsingin sanomat candidate selector (2016), `http://urn.fi/urn:nbn:fi:fsd:T-FSD2701`, version 2.1
16. Hijmans, R.J., Cameron, S.E., Parra, L.J., Jones, P.G., Jarvis, A.: Very High Resolution Interpolated Climate Surfaces for Global Land Areas. Int. J. Climatol. **25**, 1965–1978 (2005), `www.worldclim.org`
17. Johnson, A.E., Pollard, T.J., Shen, L., Lehman, L.H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L.A., Mark, R.G.: MIMIC-III, a freely accessible critical care database. Scientific data **3**, 160035 (2016)
18. Kalofolias, J., Galbrun, E., Miettinen, P.: From sets of good redescriptions to good sets of redescriptions. In: ICDM'16. pp. 211–220 (2016)
19. Kolonko, M., Wäsch, D.: Sequential reservoir sampling with a nonuniform distribution. ACM Trans. Math. Soft. **32**(2), 257–273 (2006)
20. Lee, J., Clifton, C.W.: Top-k frequent itemsets via differentially private FP-trees. In: KDD '14. pp. 931–940 (2014)
21. Malloy, T.E.: Bem sex role inventory. Wiley Online Library (2010)
22. Maruseac, M., Ghinita, G.: Precision-enhanced differentially-private mining of high-confidence association rules. IEEE Trans. Dependable Secure Comput. **17**(6), 1297–1309 (2018)
23. McSherry, F., Talwar, K.: Mechanism design via differential privacy. In: FOCS'07. pp. 94–103 (2007)
24. Mihelčić, M., Džeroski, S., Lavrač, N., Šmuc, T.: A framework for redescription set construction. Expert Syst. Appl. **68**, 196–215 (2017)
25. Mitchell-Jones, A.J., Amori, G., Bogdanowicz, W., Krystufek, B., Reijnders, P.J.H., Spitzenberger, F., Stubbe, M., Thissen, J.B.M., Vohralik, V., Zima, J.: The Atlas of European Mammals. Academic Press (1999)
26. Ramakrishnan, N., Kumar, D., Mishra, B., Potts, M., Helm, R.F.: Turning CARTwheels: An alternating algorithm for mining redescriptions. In: KDD'04. pp. 266–275 (2004)
27. Stamm, F.I., Becker, M., Strohmaier, M., Lemmerich, F.: Redescription model mining. In: KDD'21. pp. 1521–1529 (2021)
28. Tsou, Y.T., Zhen, H., Jiang, X., Huang, Y., Kuo, S.Y.: DPARM: Differentially private association rules mining. IEEE Access **8**, 142131–142147 (2020)
29. US Census Bureau: Census bureau sets key parameters to protect privacy in 2020 census results (2021), `https://www.census.gov/newsroom/press-releases/2021/2020-census-key-parameters.html`
30. Xu, S., Su, S., Xiong, L., Cheng, X., Xiao, K.: Differentially private frequent subgraph mining. In: ICDE'16. pp. 229–240 (2016)
31. Zeng, C., Naughton, J.F., Cai, J.Y.: On differentially private frequent itemset mining. Proc. VLDB Endowment **6**(1), 25–36 (2012)
32. Zinchenko, T., Galbrun, E., Miettinen, P.: Mining Predictive Redescriptions with Trees. In: ICDM'15 Workshops. pp. 1672–1675 (2015)