Privately Counting Unique Elements

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the problem of counting the number of unique elements in a dataset subject to the constraint of differential privacy. We consider the challenging setting of person-level DP (a.k.a. user-level DP) where each person may contribute an unbounded number of items and hence the sensitivity is unbounded.

Our approach is to compute a bounded-sensitivity version of this query, which reduces to solving a max-flow problem. The sensitivity bound is optimized to balance the noise we must add to privatize the answer against the error of the approximation of the bounded-sensitivity query to the true number of unique elements.

1 Introduction

An elementary data analysis task is to count the number of unique elements occurring in a dataset. The dataset may contain private data and even simple statistics can be combined to leak sensitive information about people [Dinur and Nissim, 2003]. Our goal is to release (an approximation to) this count in a way that ensures the privacy of the people who contributed their data. As a motivating example, consider a collection of internet browsing histories, in which case the goal is to compute the total number of websites that have been visited by at least one person.

Differential privacy (DP) [Dwork et al., 2006b] is a formal privacy standard. The simplest method for ensuring DP is to add noise (from either a Laplace or Gaussian distribution) to the true answer, where the scale of the noise corresponds to the sensitivity of the true answer – i.e., how much one person's data can change the true value.

If each person contributes a single element to the dataset, then the sensitivity of the number of unique elements is one. However, a person may contribute multiple elements to the dataset and our goal is to ensure privacy for all of these contributions simultaneously. That is, we seek to provide person-level DP (a.k.a. user-level DP).

This is the problem we study: We have a dataset $D = (u_1, u_2, \cdots, u_n)$ of person records. Each person $i \in [n]$ contributes a finite dataset $u_i \in \Omega^*$, where Ω is some (possibly infinite) universe of potential elements (e.g., all finite-length binary strings) and $\Omega^* := \bigcup_{\ell \in \mathbb{N}} \Omega^\ell$ denotes all subsets of Ω of finite size. Informally, our goal is to compute the number of unique elements

$$\mathrm{DC}(D) := \left| \bigcup_{i \in [n]} u_i \right| \tag{1}$$

in a way that preserves differential privacy. A priori, the sensitivity of this quantity is infinite, as a single person can contribute an unbounded number of unique elements.

In particular, it is not possible to give a meaningful upper bound on the number of distinct elements subject to differential privacy. However, it is possible to give a lower bound. Thus our formal goal is to compute a high-confidence lower bound on the number of distinct elements that is as large as possible and which is computed in a differentially private manner.

1.1 Our Contributions

Given a dataset $D = (u_1, \cdots, u_n) \in (\Omega^*)^n$ and an integer $\ell \geq 1$, we define

$$\mathrm{DC}(D; \ell) := \max \left\{ \left| \bigcup_{i \in [n]} v_i \right| : \forall i \in [n] \;\; v_i \subset u_i \wedge |v_i| \leq \ell \right\}. \tag{2}$$

That is, $\mathrm{DC}(D; \ell)$ is the number of distinct element if we restrict each person's contribution to ℓ elements. We take the maximum over all possible restrictions.

It is immediate that $\mathrm{DC}(D; \ell) \leq \mathrm{DC}(D)$ for all $\ell \geq 1$. Thus we obtain a lower bound on the true number of unique elements. The advantage of $\mathrm{DC}(D; \ell)$ is that its sensitivity is bounded by ℓ and, hence, we can estimate it in a differentially private manner. Specifically,

$$\mathcal{M}_{\ell,\varepsilon}(D) := \mathrm{DC}(D; \ell) + \mathrm{Lap}\left(\ell/\varepsilon\right)$$

defines an ε-DP algorithm $M_{\ell,\varepsilon} : (\Omega^*)^n \to \mathbb{R}$, where $\mathrm{Lap}\left(b\right)$ denotes Laplace noise scaled to have mean 0 and variance $2b^2$. This forms the basis of our algorithm. Two challenges remain: Setting the sensitivity parameter ℓ and computing $\mathrm{DC}(D; \ell)$ efficiently.

Choosing the sensitivity parameter ℓ. Any choice of $\ell \geq 1$ gives us a lower bound: $\mathrm{DC}(D; \ell) \leq \mathrm{DC}(D)$. Since $\forall D \;\lim_{\ell \to \infty} \mathrm{DC}(D; \ell) = \mathrm{DC}(D)$, this lower bound can be arbitrarily tight. However, the larger ℓ is, the larger the sensitivity of $\mathrm{DC}(D; \ell)$ is. That is, the noise we add scales linearly with ℓ.

Thus there is a bias-variance tradeoff in the choice of ℓ. To make this precise, suppose we want a lower bound on $\mathrm{DC}(D)$ with confidence $1 - \beta \in [\frac{1}{2}, 1)$. We can obtain such a lower bound from $\mathcal{M}_\ell(D)$ using the cumulative distribution function (CDF) of the Laplace distribution:

$$\begin{aligned}\mathbb{P}\left[\underbrace{\mathcal{M}_{\ell,\varepsilon}(D) - \frac{\ell}{\varepsilon} \cdot \log\left(\frac{1}{2\beta}\right)}_{\text{lower bound}} \leq \mathrm{DC}(D)\right] &= \mathbb{P}\left[\mathrm{Lap}\left(\ell/\varepsilon\right) \leq \mathrm{cdf}^{-1}_{\mathrm{Lap}(\ell/\varepsilon)}(1-\beta) + \mathrm{DC}(D) - \mathrm{DC}(D; \ell)\right] \\ &\geq \mathbb{P}\left[\mathrm{Lap}\left(\ell/\varepsilon\right) \leq \mathrm{cdf}^{-1}_{\mathrm{Lap}(\ell/\varepsilon)}(1-\beta) + 0\right] = \underbrace{1-\beta}_{\text{confidence}}.\end{aligned}$$

Thus, to obtain the tightest possible lower bound with confidence $1 - \beta$, we choose ℓ to maximize

$$q(D; \ell) := \mathrm{DC}(D; \ell) - \frac{\ell}{\varepsilon} \cdot \log\left(\frac{1}{2\beta}\right).$$

We can use the exponential mechanism [McSherry and Talwar, 2007] to privately select ℓ that approximately maximizes $q(D; \ell)$. However, directly applying the exponential mechanism is problematic because each score has a different sensitivity – the sensitivity of $q(\cdot; \ell)$ is ℓ. Instead, we apply the generalized exponential mechanism of Raskhodnikova and Smith [2015] (see Algorithm 3).

Our main algorithm attains the following guarantees.

Theorem 1.1 (Theoretical Guarantees of Our Algorithm)**.** *Let $\varepsilon > 0$ and $\beta \in (0, \frac{1}{2})$ and $\ell_{\max} \in \mathbb{N}$. Define $\mathcal{M} : (\Omega^*)^* \to \mathbb{N} \times \mathbb{R}$ to be $\mathcal{M}(D) = \textsc{DPDistinctCount}(D; \ell_{\max}, \varepsilon, \beta)$ from Algorithm 1. Then $\mathcal{M}$ satisfies all of the following properties.*

- ***Privacy:*** *$\mathcal{M}$ is ε-differentially private.*
- ***Lower bound:*** *For all $D \in (\Omega^*)^n$,*

$$\mathop{\mathbb{P}}_{(\hat{\ell}, \hat{\nu}) \leftarrow \mathcal{M}(D)}\left[\hat{\nu} \leq \mathrm{DC}(D)\right] \geq 1 - \beta. \tag{3}$$

- ***Upper bound*** *For all $D \in (\Omega^*)^n$,*

$$\mathop{\mathbb{P}}_{(\hat{\ell},\hat{\nu})\leftarrow\mathcal{M}(D)}\left[\hat{\nu} \ge \max_{\ell\in[\ell_{\max}]} \mathrm{DC}(D;\ell) - \frac{10\ell + 18\ell_A^*}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right)\right] \ge 1-2\beta, \tag{4}$$

  where $\ell_A^* = \arg\max_{\ell\in[\ell_{\max}]} \mathrm{DC}(D;\ell) - \frac{\ell}{\varepsilon}\log\left(\frac{1}{2\beta}\right)$.

- ***Computational efficiency:*** *$\mathcal{M}(D)$ has running time $O\left(|D|^{1.5}\cdot\ell_{\max}^2\right)$, where $|D| := \sum_i |u_i|$.*

In particular, if $D = (u_1, \cdots, u_n) \in (\Omega^*)^n$ satisfies $\max_{i\in[n]} |u_i| \le \ell_* \le \ell_{\max}$, then combining the upper and lower bounds of Theorem 1.1 gives

$$\mathop{\mathbb{P}}_{(\hat{\ell},\hat{\nu})\leftarrow\mathcal{M}(D)}\left[\mathrm{DC}(D) \ge \hat{\nu} \ge \mathrm{DC}(D) - \frac{28\ell_*}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right)\right] \ge 1-3\beta. \tag{5}$$

In addition to proving the above theoretical guarantees, we perform an experimental evaluation of our algorithm.

Algorithm 1 Distinct Count Algorithm

```
procedure SensitiveDistinctCount(D=(u_1,··· ,u_n)∈(Ω*)^n; ℓ∈ℕ)        ▷ DC(D; ℓ)
    Let U_ℓ = ⋃_{i∈[n]} ({i} × [min{ℓ, |u_i|}]) ⊂ [n] × [ℓ].
    Let V = ⋃_{i∈[n]} u_i ⊂ Ω.
    Define E_ℓ ⊆ U × V by ((i, j), v) ∈ E ⟺ v ∈ u_i.
    Let G_ℓ be a bipartite graph with vertices partitioned into U_ℓ and V and edges E_ℓ.
    m_ℓ ← MaximumMatchingSize(G).        ▷ [Hopcroft and Karp, 1973, Karzanov, 1973]
    return m_ℓ ∈ ℕ
end procedure
procedure DPDistinctCount(D=(u_1,··· ,u_n)∈(Ω*)^n; ℓ_max∈ℕ, ε>0, β∈(0, 1/2))
    for ℓ ∈ [ℓ_max] do
        Define q_ℓ(D) := SensitiveDistinctCount(D; ℓ) − (2ℓ/ε) · log(1/(2β)).
    end for
    ℓ̂ ← GEM(D; {q_ℓ}_{ℓ∈[ℓ_max]}, {ℓ}_{ℓ∈[ℓ_max]}, ε/2, β).        ▷ Algorithm 3
    ν̂ ← q_ℓ̂(D) + Lap(2ℓ̂/ε).
    return (ℓ̂, ν̂) ∈ [ℓ_max] × ℝ.
end procedure
```

Efficient computation. The main computational task for our algorithm is to compute $\mathrm{DC}(D;\ell)$. By definition (2), this is an optimization problem. For each person $i \in [n]$, we must select a subset v_i of that person's data u_i of size at most ℓ so as to maximize the size of the union of the subsets $\left|\bigcup_{i\in[n]} v_i\right|$.

We can view the dataset $D = (u_1, \cdots, u_n) \in (\Omega^*)^n$ as a bipartite graph. On one side we have the n people and on the other side we have the elements of the data universe Ω.[1] There is an edge between $i \in [n]$ and $x \in \Omega$ if and only if $x \in u_i$.

We can reduce computing $\mathrm{DC}(D;\ell)$ to a max-flow problem: Each edge in the bipartite graph has capacity one. We add a source vertex s which is connected to each person $i \in [n]$ by an edge with capacity ℓ. Finally we add a sink t that is connected to each $x \in \Omega$ by an edge with capacity 1. The max flow through this graph is precisely $\mathrm{DC}(D;\ell)$.

Alternatively, we can reduce computing $\mathrm{DC}(D;\ell)$ to bipartite maximum matching. For $\ell = 1$, $\mathrm{DC}(D;1)$ is exactly the maximum cardinality of a matching in the bipartite graph described above.

[1] The data universe Ω may be infinite, but we can restrict the computation to the finite set $\bigcup_{i\in[n]} u_i$. Thus there are at most $n + \mathrm{DC}(D) \le n + |D|$ item vertices in the graph.

For $\ell \geq 2$, we simply create ℓ copies of each person vertex $i \in [n]$ and then $\mathrm{DC}(D;\ell)$ is the maximum cardinality of a matching in this new bipartite graph.[2]

Using this reduction, standard algorithms for bipartite maximum matching [Hopcroft and Karp, 1973, Karzanov, 1973] allow us to compute $\mathrm{DC}(D;\ell)$ with $O(|D|^{1.5} \cdot \ell)$ operations. We must repeat this computation for each $\ell \in [\ell_{\max}]$.

Algorithm 2 Linear-Time Approximate Distinct Count Algorithm

1: **procedure** ApproxDPDistinctCount($D{=}(u_1,\cdots,u_n){\in}(\Omega^*)^n$; $\ell_{\max}{\in}\mathbb{N}$, $\varepsilon{>}0$, $\beta{\in}(0,\frac{1}{2})$)
2: $\quad S \leftarrow \emptyset$.
3: $\quad$ **for** $\ell \in [\ell_{\max}]$ **do**
4: $\quad\quad$ **for** $i \in [n]$ with $u_i \setminus S \neq \emptyset$ **do**
5: $\quad\quad\quad$ Choose lexicographically first $v \in u_i \setminus S$. $\quad\triangleright$ Match (i,ℓ) to v.
6: $\quad\quad\quad$ Update $S \leftarrow S \cup \{v\}$.
7: $\quad\quad$ **end for**
8: $\quad\quad$ Define $q_\ell(D) := |S| - \frac{2\ell}{\varepsilon}\cdot\log\left(\frac{1}{2\beta}\right)$. $\quad\triangleright$ This loop computes $\{q_\ell(D)\}_{\ell\in[\ell_{\max}]}$.
9: $\quad$ **end for**
10: $\quad \hat{\ell} \leftarrow \mathrm{GEM}(D;\{q_\ell\}_{\ell\in[\ell_{\max}]},\{\ell\}_{\ell\in[\ell_{\max}]},\varepsilon/2,\beta)$. $\quad\triangleright$ Algorithm 3
11: $\quad \hat{\nu} \leftarrow q_{\hat{\ell}}(D) + \mathrm{Lap}\left(2\hat{\ell}/\varepsilon\right)$.
12: $\quad$ **return** $(\hat{\ell},\hat{\nu}) \in [\ell_{\max}] \times \mathbb{R}$.
13: **end procedure**

Linear-time algorithm. Our algorithm above is polynomial-time. However, for many applications the dataset size $|D|$ is enormous. Thus we also propose a linear-time variant of our algorithm. However, we must trade accuracy for efficiency.

There are two key ideas that differentiate our linear-time algorithm (Algorithm 2) from our first algorithm (Algorithm 1) above: First, we compute a maxim*al* bipartite matching instead of a maxim*um* bipartite matching. This can be done using a linear-time greedy algorithm and gives a 2-approximation to the maximal matching. (Experimentally we find that the approximation is better than a factor of 2.) Second, rather than repeating the computation from scratch for each $\ell \in [\ell_{\max}]$, we incrementally update our a maximal matching while increasing ℓ. The main challenge here is ensuring that the approximation to $\mathrm{DC}(D;\ell)$ has low sensitivity – i.e., we must ensure that our approximation algorithm doesn't inflate the sensitivity. Note that $\mathrm{DC}(D;\ell)$ having low sensitivity does not automatically ensure that the approximation has low sensitivity.

Theorem 1.2 (Theoretical Guarantees of Our Linear-Time Algorithm)**.** *Let* $\varepsilon > 0$ *and* $\beta \in (0,\frac{1}{2})$ *and* $\ell_{\max} \in \mathbb{N}$. *Define* $\mathcal{M} : (\Omega^*)^* \to \mathbb{N}\times\mathbb{R}$ *to be* $\widehat{\mathcal{M}}(D) =$ ApproxDPDistinctCount$(D;\ell_{\max},\varepsilon,\beta)$ *from Algorithm 2. Then* $\widehat{\mathcal{M}}$ *satisfies all of the following properties.*

- ***Privacy:*** $\widehat{\mathcal{M}}$ *is* ε*-differentially private.*
- ***Lower bound:*** *For all* $D \in (\Omega^*)^n$,

$$\underset{(\hat{\ell},\hat{\nu})\leftarrow\widehat{\mathcal{M}}(D)}{\mathbb{P}}\left[\hat{\nu} \leq \mathrm{DC}(D)\right] \geq 1-\beta. \tag{6}$$

- ***Upper bound:*** *If* $D = (u_1,\cdots,u_n) \in (\Omega^*)^n$ *satisfies* $\max_{i\in[n]}|u_i| \leq \ell_* \leq \ell_{\max}$, *then*

$$\underset{(\hat{\ell},\hat{\nu})\leftarrow\widehat{\mathcal{M}}(D)}{\mathbb{P}}\left[\hat{\nu} \geq \frac{1}{2}\mathrm{DC}(D) - O\left(\frac{\ell_*}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right)\right)\right] \geq 1-2\beta. \tag{7}$$

- ***Computational efficiency:*** $\mathcal{M}(D)$ *has running time* $O(|D|)$, *where* $|D| := \sum_i |u_i|$.

[2] We need only create $\min\{\ell,|u_i|\}$ copies of the person $i \in [n]$. Thus the number of person vertices is at most $\min\{n\ell,|D|\}$.

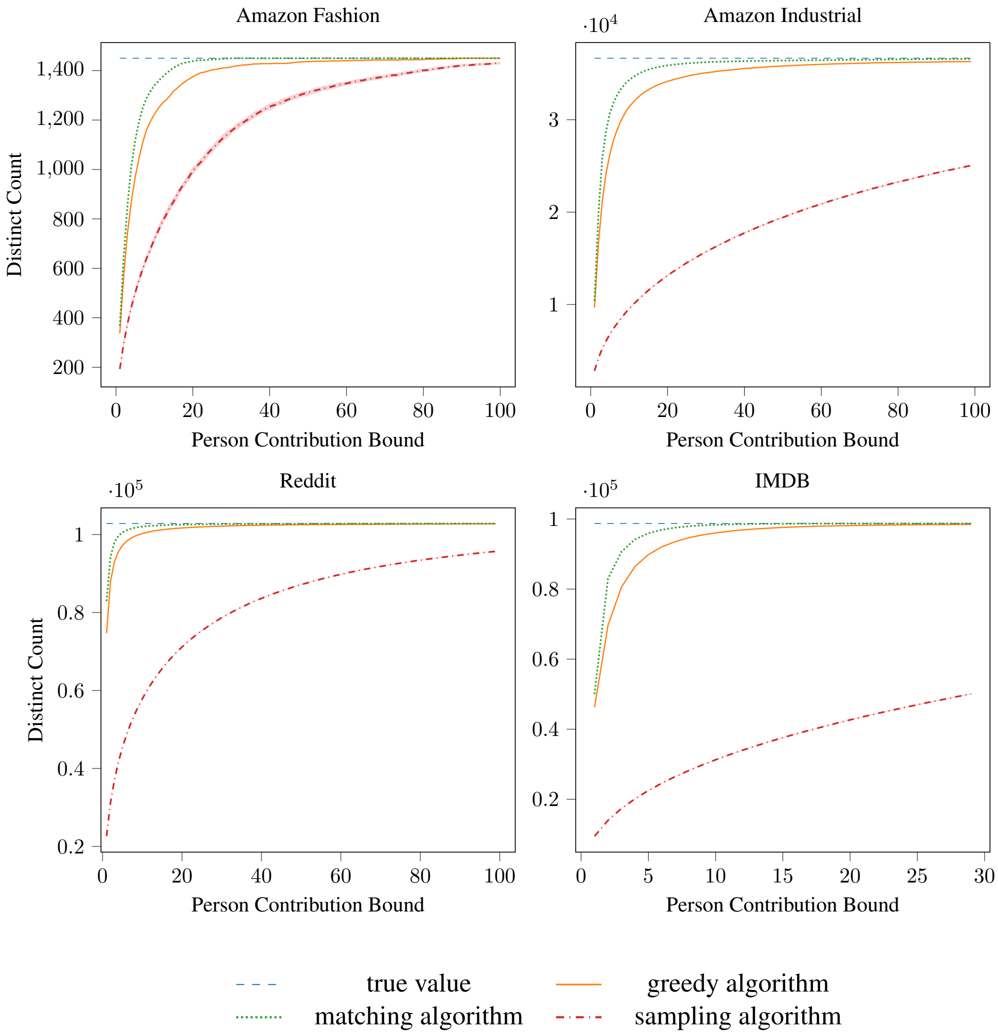

Figure 1: Performance of different algorithms estimating distinct count assuming that each person can contribute at most ℓ elements.

2 Related Work

Counting the number of distinct elements in a collection is one of the most fundamental operations. Hence, unsurprisingly, the problem of computing the number of unique elements in a differentially private way has been extensively investigated.

In the case where we assume each person contributes only one element (a.k.a. event-level privacy), the number of distinct elements has sensitivity 1 and, hence, we can simply use Laplace (or Gaussian) noise addition to release. However, it may not be possible to compute the number of distinct elements exactly (e.g. in the local model of DP [Kasiviswanathan et al., 2011]).

Most efforts have been focused on creating differentially private approximation schemes for counting distinct elements. Desfontaines et al. [2019] proved that a number of existing approximate algorithms allow an attacker to test whether a particular individual is in the collection; therefore, creation of a differentially private scheme requires care. Nonetheless, Smith et al. [2020] proved that Flajolet-Martin Sketch is private by itself and Dickens et al. [2022] proved that several other cardinality estimators can be tweaked to make them private. In case of local and shuffle models the only known results are communication complexity bounds [Chen et al., 2021]. Counting unique elements has been considered in the streaming setting [Dwork et al., 2010, Ghazi et al., 2023].

A closely related problem is that of identifying as many elements as possible (rather than just counting them); this is known as " partition selection," "set union," and "key selection" [Swanberg et al., 2023, Desfontaines et al., 2022, Korolova et al., 2009, Carvalho et al., 2022, Rivera Cardoso and Rogers, 2022, Gopi et al., 2020, Zhang et al., 2023]. Note that, by design, DP prevents us from identifying elements that only appear once in the dataset, or only a few times. Thus we can only output items that appear frequently.

For our problem of counting the number of unique elements under person-level/user-level privacy, the only known algorithm is the algorithm where each user independently samples a subset of their elements to reduce the sensitivity. We use this as a baseline in our experiments and show that our algorithm outperforms it.

3 Technical Background on Differential Privacy

For detailed background on differential privacy, see the survey by Vadhan [2017] or the book by Dwork and Roth [2014]. We briefly define pure DP and some basic mechanisms and results.

Algorithm 3 Generalized Exponential Mechanism [Raskhodnikova and Smith, 2015]

procedure GEM($D \in \mathcal{X}^*$; $q_i : \mathcal{X}^* \to \mathbb{R}$ for $i \in [m]$, $\Delta_i > 0$ for $i \in [m]$, $\varepsilon > 0$, $\beta > 0$)
  Require: q_i has sensitivity $\sup_{\substack{x,x' \in \mathcal{X}^* \\ \text{neighboring}}} |q(x) - q(x')| \le \Delta_i$ for all $i \in [m]$.
  Let $t = \frac{2}{\varepsilon} \log\left(\frac{m}{\beta}\right)$.
  for $i \in [m]$ **do**
    $s_i \leftarrow \min_{j \in [m]} \frac{(q_i(D) - t\Delta_i) - (q_j(D) - t\Delta_j)}{\Delta_i + \Delta_j}$.
  end for
  Sample $\hat{i} \in [m]$ from the Exponential Mechanism using the normalized scores s_i; i.e.,

$$\forall i \in [m] \qquad \mathbb{P}\left[\hat{i} = i\right] = \frac{\exp\left(\frac{1}{2}\varepsilon s_i\right)}{\sum_{k \in [m]} \exp\left(\frac{1}{2}\varepsilon s_k\right)}.$$

  return $\hat{i} \in [m]$.
end procedure

Definition 3.1 (Differential Privacy (DP) [Dwork et al., 2006b]). *A randomized algorithm $M : \mathcal{X}^* \to \mathcal{Y}$ satisfies ε-DP if, for all inputs $D, D' \in \mathcal{X}^*$ differing only by the addition or removal of an element and for all measurable $S \subset \mathcal{Y}$, we have $\mathbb{P}[M(D) \in S] \le e^{\varepsilon} \cdot \mathbb{P}[M(D') \in S]$.*

We refer to pairs of inputs that differ only by the addition or removal of one person's data as *neighboring*. Note that it is common to also consider replacement of one person's data; for simplicity, we do not do this. We remark that there are also variants of DP such as approximate DP [Dwork et al., 2006a] and concentrated DP [Dwork and Rothblum, 2016, Bun and Steinke, 2016], which quantitatively relax the definition, but these are not relevant in our application. A key property of DP is that it composes and is invariant under postprocessing.

Lemma 3.2 (Composition & Postprocessing). *Let $M_1 : \mathcal{X}^* \to \mathcal{Y}$ be ε_1-DP. Let $M_2 : \mathcal{X}^* \times \mathcal{Y} \to \mathcal{Z}$ be such that, for all $y \in \mathcal{Y}$, the restriction $M(\cdot, y) : \mathcal{X}^* \to \mathcal{Z}$ is ε_2-DP. Define $M_{12} : \mathcal{X}^* \to \mathcal{Z}$ by $M_{12}(D) = M_2(D, M_1(D))$. Then M_{12} is $(\varepsilon_1 + \varepsilon_2)$-DP.*

A basic DP tool is the Laplace mechanism [Dwork et al., 2006b]. Note that we could also use the *discrete* Laplace mechanism [Ghosh et al., 2009, Canonne et al., 2020].

Lemma 3.3 (Laplace Mechanism). *Let $q : \mathcal{X}^* \to \mathbb{R}$. We say q has sensitivity Δ if $|q(D) - q(D')| \le \Delta$ for all neighboring $D, D' \in \mathcal{X}^*$. Define $M : \mathcal{X}^* \to \mathbb{R}$ by $M(D) = q(D) + \mathrm{Lap}(\Delta/\varepsilon)$, where $\mathrm{Lap}(b)$ denotes laplace noise with mean 0 and variance $2b^2$ – i.e., $\underset{\xi \leftarrow \mathrm{Lap}(b)}{\mathbb{P}}[\xi > t] = \underset{\xi \leftarrow \mathrm{Lap}(b)}{\mathbb{P}}[\xi < -t] = \frac{1}{2}\exp\left(\frac{t}{b}\right)$ for all $t > 0$. Then M is ε-DP.*

Another fundamental tool for DP is the exponential mechanism [McSherry and Talwar, 2007]. It selects the approximately best option from among a set of options, where each option i has a quality function q_i with sensitivity Δ. The following result generalizes the exponential mechanism by allowing each of the quality functions to have a different sensitivity.

Data Set	Size		Words per Person			Vocabulary Size
	People	Records	Min	Median	Max	
Amazon Fashion	404	8533	1	14.0	139	1450
Amazon Industrial and Scientific	11041	1446031	0	86	2059	36665
Reddit	223388	7117494	0	18.0	1724	102835
IMDB	50000	6688844	5	110.0	925	98726

Table 1: Data sets details.

Theorem 3.4 (Generalized Exponential Mechanism [Raskhodnikova and Smith, 2015, Theorem 1.4])**.** *For each $i \in [m]$, let $q_i : \mathcal{X}^* \to \mathbb{R}$ be a query with sensitivity Δ_i. Let $\varepsilon, \beta > 0$. The generalized exponential mechanism ($\mathrm{GEM}(\cdot; \{q_i\}_{i\in[m]}, \{\Delta_i\}_{i\in[m]}, \varepsilon, \beta)$ in Algorithm 3) is ε-DP and has the following utility guarantee. For all $D \in \mathcal{X}^*$, we have*

$$\underset{\hat{i}\leftarrow \mathrm{GEM}(D;\{q_i\}_{i\in[m]},\{\Delta_i\}_{i\in[m]},\varepsilon,\beta)}{\mathbb{P}} \left[q_{\hat{i}}(D) \geq \max_{j\in[m]} q_j(D) - \Delta_j \cdot \frac{4}{\varepsilon} \log\left(\frac{m}{\beta}\right) \right] \geq 1-\beta.$$

4 Experimental Results

We empirically validate the performance of our algorithms using data sets of various sizes from different text domains. We focus on the problem of computing vocabulary size with person-level DP. Section 4.1 describes the data sets and Section 4.2 discusses the algorithms we compare.

4.1 Datasets

We used four publicly available datasets to assess the accuracy of our algorithms compared to baselines. Two small datasets were used: Amazon Fashion 5-core [Ni et al., 2019] (reviews of fashion products on Amazon) and Amazon Industrial and Scientific 5-core [Ni et al., 2019] (reviews of industrial and scientific products on Amazon). Two large data sets were also used: Reddit [Shen, 2020] (a data set of posts collected from r/AskReddit) and IMDb [N, 2020, Maas et al., 2011] (a set of movie reviews scraped from IMDb). See details of the datasets in Table 1.

4.2 Comparisons

Computing the number of distinct elements using a differentially private mechanism involves two steps: selecting a contribution bound (ℓ in our algorithms) and counting the number of distinct elements in a way that restricts each person to only contribute the given number of elements.

Selection: We examine three algorithms for determining the contribution limit:

1. Choosing the true maximum person contribution (due to computational restrictions this was only computed for Amazon Fashion data set).
2. Choosing the 90th percentile of person contributions.
3. Choosing the person contribution that maximizes the utility function $q_\ell(D) = \mathrm{DC}(D;\ell) - \frac{\ell}{\varepsilon}\log(\frac{1}{2\beta})$, where $\varepsilon = 1$, and $\beta = 0.001$.
4. Choosing the person contribution that maximizes the utility function using generalized exponential mechanism with $\epsilon = 1$.

Note that only the last option is differentially private, but we consider the other comparison points nonetheless.

Counting: We also consider three algorithms for estimating the number of distinct elements for a given sensitivity bound ℓ:

1. For each person, we independently sample ℓ elements and count the number of distinct elements in the union of the samples.

Selection	Counting	Person Contribution Bound			Distinct Count		
		10th PC	Median	90th PC	10th PC	Median	90th PC
Max Contrib	DP Sampling	–	139	–	1196.8	1407.5	1649.1
Max Contrib	DP Greedy	–	139	–	1174.2	1439.2	1646.5
Max Contrib	DP Matching	–	139	–	1222.2	1460.9	1631.0
90th PC Contrib	DP Sampling	–	48	–	1225.4	1296.2	1377.9
90th PC Contrib	DP Greedy	–	48	–	1367.0	1432.6	1516.3
90th PC Contrib	DP Matching	–	48	–	1365.3	1444.7	1524.8
Max Utility	Sampling	–	41	–	1247.0	1259.0	1270.0
Max Utility	Greedy	–	20	–	–	1376	–
Max Utility	Matching	–	17	–	–	1428	–
DP Max Utility	Sampling	8.9	16.0	28.0	661.6	892.5	1124.5
DP Max Utility	Greedy	8.0	11.0	17.0	1148.0	1241.0	1348.0
DP Max Utility	Matching	7.0	9.0	14.0	1252.0	1317.0	1400.0
DP Max Utility	DP Sampling	9.0	16.0	27.1	702.4	899.1	1145.1
DP Max Utility	DP Greedy	8.0	10.0	19.0	1128.5	1224.4	1370.8
DP Max Utility	DP Matching	6.9	9.0	13.1	1220.6	1319.1	1394.2

Table 2: Amazon Fashion: the comparison is for $\ell_{\max} = 100$.

Selection	Counting	Person Contribution Bound			Distinct Count		
		10th PC	Median	90th PC	10th PC	Median	90th PC
90th PC Contrib	DP Sampling	–	297	–	32458.1	32943.8	33452.6
90th PC Contrib	DP Greedy	–	297	–	36270.3	36669.5	37019.0
90th PC Contrib	DP Matching	–	297	–	36236.2	36651.7	37102.7
Max Utility	Sampling	–	99	–	24967.0	25039.0	25121.2
Max Utility	Greedy	–	79	–	–	36246	–
Max Utility	Matching	–	42	–	–	36364	–
DP Max Utility	Sampling	85.9	96.0	99.0	23852.8	24739.0	25049.8
DP Max Utility	Greedy	34.0	49.0	66.1	35393.0	35839.0	36116.9
DP Max Utility	Matching	22.9	30.5	43.2	36026.8	36243.5	36371.2
DP Max Utility	DP Sampling	87.0	95.0	99.0	23997.6	24701.1	25067.7
DP Max Utility	DP Greedy	32.9	47.5	68.0	35336.6	35776.2	36136.6
DP Max Utility	DP Matching	22.0	28.0	38.0	35970.5	36198.9	36326.7

Table 3: Amazon Industrial and Scientific: the comparison is for $\ell_{\max} = 100$.

2. The linear-time greedy algorithm (Algorithm 2) with $\varepsilon = 1$ and $\beta = 0.001$.
3. The matching-based algorithm (Algorithm 1) with $\varepsilon = 1$ and $\beta = 0.001$.

All of these can be converted into DP algorithms by adding Laplace noise to the result.

In all our datasets "true maximum person contribution" and "90th percentile of person contributions" output bounds that are much larger than necessary to obtain true distinct count; hence, we only consider DP versions of the estimation algorithm for these selection algorithms.

4.3 Results

Figure 1 shows the dependency of the result on the contribution bound for each of the algorithms for computing the number of distinct elements with fixed person contribution. It is clear that matching and greedy algorithms vastly outperform the sampling approach that is currently used in practice.

Tables 2 to 5 show the performance of algorithms for selecting optimal person contribution bounds on different data sets. For all bound selection algorithms and all data sets, the sampling approach to

Selection	Counting	Person Contribution Bound			Distinct Count		
		10th PC	Median	90th PC	10th PC	Median	90th PC
90th PC Contrib	DP Sampling	–	75	–	92480.7	92654.8	92812.1
90th PC Contrib	DP Greedy	–	75	–	102544.8	102665.7	102817.7
90th PC Contrib	DP Matching	–	75	–	102651.1	102784.1	102907.8
Max Utility	Sampling	–	99	–	95606.9	95692.0	95750.3
Max Utility	Greedy	–	52	–	–	102543	–
Max Utility	Matching	–	32	–	–	102685	–
DP Max Utility	Sampling	89.0	96.0	99.0	94549.9	95394.5	95656.5
DP Max Utility	Greedy	26.0	33.0	50.0	102015.0	102253.0	102527.0
DP Max Utility	Matching	14.0	18.5	30.0	102357.0	102501.5	102671.0
DP Max Utility	DP Sampling	88.8	96.0	99.0	94665.2	95375.5	95693.5
DP Max Utility	DP Greedy	27.0	34.0	53.0	102053.2	102289.6	102531.2
DP Max Utility	DP Matching	14.9	18.5	28.0	102379.7	102512.6	102643.9

Table 4: Reddit: the comparison is for $\ell_{\max} = 100$.

Selection	Counting	Person Contribution Bound			Distinct Count		
		10th PC	Median	90th PC	10th PC	Median	90th PC
90th PC Contrib	DP Sampling	–	238	–	95264.5	95593.5	95966.1
90th PC Contrib	DP Greedy	–	238	–	98411.0	98734.0	99120.0
90th PC Contrib	DP Matching	–	238	–	98354.2	98729.4	99164.2
Max Utility	Sampling	–	29	–	49907.8	50036.5	50195.3
Max Utility	Greedy	–	29	–	–	98459	–
Max Utility	Matching	–	19	–	–	98712	–
DP Max Utility	Sampling	29.0	29.0	29.0	49899.6	50070.5	50220.9
DP Max Utility	Greedy	22.0	25.0	29.0	98244.0	98364.0	98459.0
DP Max Utility	Matching	13.0	16.0	21.0	98586.0	98674.0	98721.0
DP Max Utility	DP Sampling	29.0	29.0	29.0	49924.2	50053.7	50211.9
DP Max Utility	DP Greedy	20.0	26.0	29.0	98126.7	98369.6	98451.8
DP Max Utility	DP Matching	12.0	16.0	21.0	98555.6	98670.4	98726.8

Table 5: IMDB: the comparison is for $\ell_{\max} = 30$.

estimating the distinct count performs much worse than the greedy and matching-based approaches. The greedy approach performs worse than the matching-based approach, but the difference is about 10% for Amazon Fashion and is almost negligible for other data sets since they are much larger. As for the matching-based algorithm, it performs as follows on all the data sets:

1. The algorithm that uses the bound equal to the maximal person contribution overestimates the actual necessary bound. Therefore, we only consider the DP algorithms for counts estimation. It is easy to see that while the median of the estimation is close to the actual distinct count, the amount of noise is somewhat large.
2. The algorithm that uses the bound equal to the 99th percentile of person contributions also overestimates the necessary bound and behaves similarly to the one we just described (though the spread of the noise is a bit smaller).
3. The algorithms that optimize the utility function are considered: one non-private and one private. The non-private algorithm with non-private estimation gives the answer that is very close to the true number of distinct elements. The private algorithm with non-private estimation gives the answer that is worse, but not too much. Finally, the private algorithm with the private estimation gives answers very similar to the results of the non-private estimation.

References

Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography: 14th International Conference, TCC 2016-B, Beijing, China, October 31-November 3, 2016, Proceedings, Part I*, pages 635–658. Springer, 2016. URL https://arxiv.org/abs/1605.02065.

Clément L Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *Advances in Neural Information Processing Systems*, 33:15676–15688, 2020. URL https://arxiv.org/abs/2004.00010.

Ricardo Silva Carvalho, Ke Wang, and Lovedeep Singh Gondara. Incorporating item frequency for differentially private set union. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 9504–9511. AAAI Press, 2022. URL https://ojs.aaai.org/index.php/AAAI/article/view/21183.

Lijie Chen, Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. On distributed differential privacy and counting distinct elements. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 56:1–56:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi: 10.4230/LIPIcs.ITCS.2021.56. URL https://doi.org/10.4230/LIPIcs.ITCS.2021.56.

Damien Desfontaines, Andreas Lochbihler, and David A. Basin. Cardinality estimators do not preserve privacy. *Proc. Priv. Enhancing Technol.*, 2019(2):26–46, 2019. doi: 10.2478/popets-2019-0018. URL https://doi.org/10.2478/popets-2019-0018.

Damien Desfontaines, James Voss, Bryant Gipson, and Chinmoy Mandayam. Differentially private partition selection. *Proc. Priv. Enhancing Technol.*, 2022(1):339–352, 2022. doi: 10.2478/popets-2022-0017. URL https://doi.org/10.2478/popets-2022-0017.

Charlie Dickens, Justin Thaler, and Daniel Ting. Order-invariant cardinality estimators are differentially private. In *NeurIPS*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/623307df18da128262aaf394cdcfb235-Abstract-Conference.html.

Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. URL https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf.

Cynthia Dwork and Guy N Rothblum. Concentrated differential privacy. *arXiv preprint arXiv:1603.01887*, 2016. URL https://arxiv.org/abs/1603.01887.

Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*, pages 486–503. Springer, 2006a.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006b.

Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N. Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 66–80. Tsinghua University Press, 2010. URL http://conference.iiis.tsinghua.edu.cn/ICS2010/content/papers/6.html.

Badih Ghazi, Ravi Kumar, Jelani Nelson, and Pasin Manurangsi. Private counting of distinct and k-occurring items in time windows. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 55:1–55:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi: 10.4230/LIPIcs.ITCS.2023.55. URL https://doi.org/10.4230/LIPIcs.ITCS.2023.55.

Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 351–360, 2009. URL https://arxiv.org/abs/0811.2841.

Sivakanth Gopi, Pankaj Gulhane, Janardhan Kulkarni, Judy Hanwen Shen, Milad Shokouhi, and Sergey Yekhanin. Differentially private set union. In *International Conference on Machine Learning*, pages 3627–3636. PMLR, 2020. URL https://arxiv.org/abs/2002.09745.

John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973. doi: 10.1137/0202019. URL https://doi.org/10.1137/0202019.

A.V. Karzanov. An exact estimate of an algorithm for finding a maximum flow, applied to the problem "on representatives". *Issues of Cybernetics. Proc. of the Seminar on Combinatorial Mathematics*, pages 66–70, 1973.

Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011. URL https://arxiv.org/abs/0803.0924.

Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. Releasing search queries and clicks privately. In Juan Quemada, Gonzalo León, Yoëlle S. Maarek, and Wolfgang Nejdl, editors, *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 171–180. ACM, 2009. doi: 10.1145/1526709.1526733. URL https://doi.org/10.1145/1526709.1526733.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

Lakshmipathi N. Imdb dataset of 50k movie reviews, 2020. URL https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews.

Jianmo Ni, Jiacheng Li, and Julian J. McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 188–197. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1018. URL https://doi.org/10.18653/v1/D19-1018.

Sofya Raskhodnikova and Adam D. Smith. Efficient lipschitz extensions for high-dimensional graph statistics and node private degree distributions. *CoRR*, abs/1504.07912, 2015. URL http://arxiv.org/abs/1504.07912.

Adrian Rivera Cardoso and Ryan Rogers. Differentially private histograms under continual observation: Streaming selection into the unknown. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 2397–2419. PMLR, 28–30 Mar 2022. URL https://arxiv.org/abs/2103.16787.

Judy Hanwen Shen. Ask reddit, 2020. URL https://github.com/heyyjudes/differentially-private-set-union/tree/ea7b39285dace35cc9e9029692802759f3e1c8e8/data.

Adam D. Smith, Shuang Song, and Abhradeep Thakurta. The flajolet-martin sketch itself preserves differential privacy: Private counting with minimal space. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/e3019767b1b23f82883c9850356b71d6-Abstract.html.

Marika Swanberg, Damien Desfontaines, and Samuel Haney. DP-SIPS: A simpler, more scalable mechanism for differentially private partition selection. *CoRR*, abs/2301.01998, 2023. doi: 10.48550/arXiv.2301.01998. URL https://doi.org/10.48550/arXiv.2301.01998.

Salil Vadhan. The complexity of differential privacy. *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, pages 347–450, 2017. URL https://privacytools.seas.harvard.edu/files/privacytools/files/manuscript_2016.pdf.

Bing Zhang, Vadym Doroshenko, Peter Kairouz, Thomas Steinke, Abhradeep Thakurta, Ziyin Ma, Himani Apte, and Jodi Spacek. Differentially private stream processing at scale. *arXiv preprint arXiv:2303.18086*, 2023. URL https://arxiv.org/abs/2303.18086.

A Proofs

Proof of Theorem 1.1. First note that $q_\ell(D) = \mathrm{DC}(D;\ell) - \frac{2\ell}{\varepsilon}\log(1/2\beta)$ has sensitivity ℓ. Since the generalized exponential mechanism is $\varepsilon/2$-DP and adding Laplace noise is also $\varepsilon/2$-DP, the overall algorithm is ε-DP by composition.

Since $\hat{\nu} \leftarrow q_{\hat{\ell}}(D) + \mathrm{Lap}\left(2\hat{\ell}/\varepsilon\right)$, we have

$$\underset{\hat{\nu}}{\mathbb{P}}\left[\hat{\nu} \leq q_{\hat{\ell}}(D) + \frac{2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right)\right] = \underset{\hat{\nu}}{\mathbb{P}}\left[\hat{\nu} \geq q_{\hat{\ell}}(D) - \frac{2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right)\right] = 1-\beta. \tag{8}$$

Substituting $q_\ell(D) = \mathrm{DC}(D;\ell) - \frac{2\ell}{\varepsilon}\log(1/2\beta)$ into Equation (8) gives

$$\underset{\hat{\nu}}{\mathbb{P}}\left[\hat{\nu} \leq \mathrm{DC}(D;\hat{\ell})\right] = \tag{9}$$

$$\underset{\hat{\nu}}{\mathbb{P}}\left[\hat{\nu} \geq \mathrm{DC}(D;\hat{\ell}) - \frac{4\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right)\right] = 1-\beta. \tag{10}$$

Combining Equation (9) with $\mathrm{DC}(D;\hat{\ell}) \leq \mathrm{DC}(D)$ yields the guarantee in Equation (3) that $\hat{\nu}$ is a lower bound on $\mathrm{DC}(D)$ with probability $\geq 1-\beta$.

The accuracy guarantee of the generalized exponential mechanism (Theorem 3.4) is

$$\underset{\hat{\ell}}{\mathbb{P}}\left[q_{\hat{\ell}}(D) \geq \max_{\ell\in[\ell_{\max}]} q_\ell(D) - \ell\cdot\frac{4}{\varepsilon/2}\log(\ell_{\max}/\beta)\right] \geq 1-\beta$$

or, equivalently,

$$\underset{\hat{\ell}}{\mathbb{P}}\left[\mathrm{DC}(D;\hat{\ell}) - \frac{2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right) \geq \max_{\ell\in[\ell_{\max}]} \mathrm{DC}(D;\ell) - \frac{2\ell}{\varepsilon}\log\left(\frac{1}{2\beta}\right) - \frac{8\ell}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right)\right] \geq 1-\beta. \tag{11}$$

Combining Equations (10) and (11) with a union bound yields

$$\underset{(\hat{\ell},\hat{\nu})\leftarrow\mathcal{M}(D)}{\mathbb{P}}\left[\hat{\nu} \geq \max_{\ell\in[\ell_{\max}]} \mathrm{DC}(D;\ell) - \frac{2\ell+2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right) - \frac{8\ell}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right)\right] \geq 1-2\beta. \tag{12}$$

To interpret Equation (12) we need a high-probability upper bound on $\hat{\ell}$. Let $A > 0$ be determined later and define

$$\ell_A^* := \underset{\ell \in [\ell_{\max}]}{\arg\max}\ \mathrm{DC}(D;\ell) - \frac{A\ell}{\varepsilon}, \tag{13}$$

so that $\mathrm{DC}(D;\ell) \le \mathrm{DC}(D;\ell_A^*) + (\ell - \ell_A^*)\frac{A}{\varepsilon}$ for all $\ell \in [\ell_{\max}]$. Assume the event in Equation (11) holds. We have

$$\mathrm{DC}(D;\hat{\ell}) - \frac{2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right) \le \mathrm{DC}(D;\ell_A^*) + (\hat{\ell} - \ell_A^*)\frac{A}{\varepsilon} - \frac{2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right), \quad \text{(by Equation (13))}$$

$$\mathrm{DC}(D;\hat{\ell}) - \frac{2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right) \ge \max_{\ell \in [\ell_{\max}]} \mathrm{DC}(D;\ell) - \frac{2\ell}{\varepsilon}\log\left(\frac{1}{2\beta}\right) - \frac{8\ell}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right) \quad \text{(by assumption)}$$

$$\ge \mathrm{DC}(D;\ell_A^*) - \frac{2\ell_A^*}{\varepsilon}\log\left(\frac{1}{2\beta}\right) - \frac{8\ell_A^*}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right).$$

Combining inequalities and simplifying yields

$$\hat{\ell}\cdot\left(2\log\left(\frac{1}{2\beta}\right) - A\right) \le \ell_A^*\cdot\left(2\log\left(\frac{1}{2\beta}\right) + 8\log\left(\frac{\ell_{\max}}{\beta}\right) - A\right). \tag{14}$$

Now we set $A = \log\left(\frac{1}{2\beta}\right)$ to obtain

$$\hat{\ell}\cdot\log\left(\frac{1}{2\beta}\right) \le \ell_A^*\cdot\left(\log\left(\frac{1}{2\beta}\right) + 8\log\left(\frac{\ell_{\max}}{\beta}\right)\right). \tag{15}$$

Substituting Equation (15) into Equation (12) gives

$$\underset{(\hat{\ell},\hat{\nu})\leftarrow\mathcal{M}(D)}{\mathbb{P}}\left[\hat{\nu} \ge \max_{\ell\in[\ell_{\max}]} \mathrm{DC}(D;\ell) - \frac{2\ell + 2\ell_A^*}{\varepsilon}\log\left(\frac{1}{2\beta}\right) - \frac{8\ell + 16\ell_A^*}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right)\right] \ge 1 - 2\beta. \tag{16}$$

We simplify Equation (16) using $\log\left(\frac{1}{2\beta}\right) \le \log\left(\frac{\ell_{\max}}{\beta}\right)$ to obtain Equation (4).

Finally, the runtime of DPDistinctCount(D) is dominated by $\ell_{\max}$ calls to the SensitiveDistinctCount(D) subroutine, which computes the maximum size of a bipartite matching on a graph with $|E| = \sum_{i\in[n]}|u_i|\cdot\min\{\ell,|u_i|\} \le |D|\cdot\ell_{\max}$ edges and $|V|+|U| = \mathrm{DC}(D) + \sum_{i\in[n]}\min\{\ell,|u_i|\} \le 2|D|$ vertices. The Hopcroft-Karp-Karzanov algorithm runs in time $O(|E|\cdot\sqrt{|V|+|U|}) \le O(|D|^{1.5}\cdot\ell_{\max})$ time. □

Proof of Theorem 1.2. We start from proving privacy guarantees. Note that Algorithm 2 produces the same result as Algorithm 4. Hence, it is enough to prove that SensitiveApproxDistinctCount$(\cdot,\ell)$ has sensitivity ℓ. In addition, note that

$$\textsc{SensitiveApproxDistinctCount}((u_1,\cdots,u_n),\ell) = \textsc{SensitiveApproxDistinctCount}((\underbrace{u_1,\cdots,u_n,\cdots,u_1,\cdots,u_n}_{\ell\text{ times}}),1);$$

therefore, it is enough to prove that SensitiveApproxDistinctCount$(\cdot,1)$ has sensitivity 1.

Assume $D' = (u_1,\cdot,u_{j-1},u_{j+1},\ldots,u_n)$ and let $S_1,\ldots,S_n$, $v_1,\ldots,v_n$ be states of S and v ($v_i = \perp$ if i is skipped), respectively, when run ApproxSensitiveDistinctCount(D) and $S'_1,\ldots,S'_n$, $v'_1,\ldots,v'_n$ be states of S and v ($v_i = \perp$ if i is skipped), respectively, when run ApproxSensitiveDistinctCount(D'). Let $\{i_1,\ldots,i_k\} = \{i : S_i \ne S'_i\}$. It is clear that $i_1 \ge j$ and $v'_{i_1} = v_j$; similarly v'_{i_2} is either $\perp$ or $v'_{i_2} = v_{i_1}$ etc. As a result $|S'_n| \le |S_n| \le |S'_n| + 1$.

The sensitivity bound implies that $q_\ell(D) = |S| - \frac{2\ell}{\varepsilon}\log(1/2\beta)$ has sensitivity ℓ. Since the generalized exponential mechanism is $\varepsilon/2$-DP and adding Laplace noise is also $\varepsilon/2$-DP, the overall algorithm is ε-DP by composition.

Algorithm 4 Approximate Distinct Count Algorithm

procedure SensitiveApproxDistinctCount($D = (u_1, \cdots, u_n) \in (\Omega^*)^n$; $\ell \in \mathbb{N}$)
  $S \leftarrow \emptyset$.
  for $\ell' \in [\ell]$ **do**
    for $i \in [n]$ with $u_i \setminus S \neq \emptyset$ **do**
      Choose lexicographically first $v \in u_i \setminus S$. ▷ Match (i, ℓ) to v.
      Update $S \leftarrow S \cup \{v\}$.
    end for
  end for
  return $|S|$
end procedure
procedure DPApproxDistinctCount($D = (u_1, \cdots, u_n) \in (\Omega^*)^n$; $\ell_{\max} \in \mathbb{N}$, $\varepsilon > 0$, $\beta \in (0, \frac{1}{2})$)
  for $\ell \in [\ell_{\max}]$ **do**
    Define $q_\ell(D) :=$ SensitiveApproxDistinctCount$(D; \ell) - \frac{2\ell}{\varepsilon} \cdot \log\left(\frac{1}{2\beta}\right)$.
  end for
  $\hat{\ell} \leftarrow \mathrm{GEM}(D; \{q_\ell\}_{\ell \in [\ell_{\max}]}, \{\ell\}_{\ell \in [\ell_{\max}]}, \varepsilon/2, \beta)$. ▷ Algorithm 3
  $\hat{\nu} \leftarrow q_{\hat{\ell}}(D) + \mathrm{Lap}\left(2\hat{\ell}/\varepsilon\right)$.
  return $(\hat{\ell}, \hat{\nu}) \in [\ell_{\max}] \times \mathbb{R}$.
end procedure

Let us denote by $\widehat{\mathrm{DC}}(D; \ell)$ the value of $|S|$ we obtain on Line 8 in Algorithm 2. Note that $\widehat{\mathrm{DC}}(D; \ell)$ is size of a maximal matching in G_ℓ, where G_ℓ is the bipartite graph corresponding to the input D with ℓ copies of each person (see Algorithm 1 for a formal description of the graph). Since a maximal matching is a 2-approximation to a maximum matching, we have $\widehat{\mathrm{DC}}(D; \ell) \geq \frac{1}{2}\mathrm{DC}(D; \ell)$. Also $\widehat{\mathrm{DC}}(D; \ell) \leq \mathrm{DC}(D; \ell)$.

Since $\hat{\nu} \leftarrow q_{\hat{\ell}}(D) + \mathrm{Lap}\left(2\hat{\ell}/\varepsilon\right)$, we have

$$\underset{\hat{\nu}}{\mathbb{P}}\left[\hat{\nu} \leq q_{\hat{\ell}}(D) + \frac{2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right)\right] = \underset{\hat{\nu}}{\mathbb{P}}\left[\hat{\nu} \geq q_{\hat{\ell}}(D) - \frac{2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right)\right] = 1 - \beta. \tag{17}$$

Substituting $q_\ell(D) = \widehat{\mathrm{DC}}(D; \ell) - \frac{2\ell}{\varepsilon}\log(1/2\beta)$ into Equation (17) gives

$$\underset{\hat{\nu}}{\mathbb{P}}\left[\hat{\nu} \leq \widehat{\mathrm{DC}}(D; \hat{\ell})\right] = \tag{18}$$

$$\underset{\hat{\nu}}{\mathbb{P}}\left[\hat{\nu} \geq \widehat{\mathrm{DC}}(D; \hat{\ell}) - \frac{4\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right)\right] = 1 - \beta. \tag{19}$$

Combining Equation (18) with $\widehat{\mathrm{DC}}(D; \hat{\ell}) \leq \mathrm{DC}(D)$ yields the guarantee in Equation (6) that $\hat{\nu}$ is a lower bound on $\mathrm{DC}(D)$ with probability $\geq 1 - \beta$.

The accuracy guarantee of the generalized exponential mechanism (Theorem 3.4) is

$$\underset{\hat{\ell}}{\mathbb{P}}\left[q_{\hat{\ell}}(D) \geq \max_{\ell \in [\ell_{\max}]} q_\ell(D) - \ell \cdot \frac{4}{\varepsilon/2}\log(\ell_{\max}/\beta)\right] \geq 1 - \beta$$

or, equivalently,

$$\underset{\hat{\ell}}{\mathbb{P}}\left[\widehat{\mathrm{DC}}(D; \hat{\ell}) - \frac{2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right) \geq \max_{\ell \in [\ell_{\max}]} \widehat{\mathrm{DC}}(D; \ell) - \frac{2\ell}{\varepsilon}\log\left(\frac{1}{2\beta}\right) - \frac{8\ell}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right)\right] \geq 1 - \beta. \tag{20}$$

Combining Equations (19) and (20) with a union bound yields

$$\underset{(\hat{\ell}, \hat{\nu}) \leftarrow \mathcal{M}(D)}{\mathbb{P}}\left[\hat{\nu} \geq \max_{\ell \in [\ell_{\max}]} \widehat{\mathrm{DC}}(D; \ell) - \frac{2\ell + 2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right) - \frac{8\ell}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right)\right] \geq 1 - 2\beta. \tag{21}$$

To interpret Equation (21) we need a high-probability upper bound on $\hat{\ell}$. Let $A > 0$ be determined later and define

$$\ell_A^* := \underset{\ell \in [\ell_{\max}]}{\arg\max}\ \widehat{\mathrm{DC}}(D;\ell) - \frac{A\ell}{\varepsilon}, \tag{22}$$

so that $\widehat{\mathrm{DC}}(D;\ell) \le \widehat{\mathrm{DC}}(D;\ell_A^*) + (\ell - \ell_A^*)\frac{A}{\varepsilon}$ for all $\ell \in [\ell_{\max}]$. Assume the event in Equation (20) holds. We have

$$\widehat{\mathrm{DC}}(D;\hat{\ell}) - \frac{2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right) \le \widehat{\mathrm{DC}}(D;\ell_A^*) + (\hat{\ell} - \ell_A^*)\frac{A}{\varepsilon} - \frac{2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right), \qquad \text{(by Equation (22))}$$

$$\begin{aligned}\widehat{\mathrm{DC}}(D;\hat{\ell}) - \frac{2\hat{\ell}}{\varepsilon}\log\left(\frac{1}{2\beta}\right) &\ge \max_{\ell \in [\ell_{\max}]} \widehat{\mathrm{DC}}(D;\ell) - \frac{2\ell}{\varepsilon}\log\left(\frac{1}{2\beta}\right) - \frac{8\ell}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right) \\ &\hspace{20em}\text{(by assumption)} \\ &\ge \widehat{\mathrm{DC}}(D;\ell_A^*) - \frac{2\ell_A^*}{\varepsilon}\log\left(\frac{1}{2\beta}\right) - \frac{8\ell_A^*}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right).\end{aligned}$$

Combining inequalities and simplifying yields

$$\hat{\ell} \cdot \left(2\log\left(\frac{1}{2\beta}\right) - A\right) \le \ell_A^* \cdot \left(2\log\left(\frac{1}{2\beta}\right) + 8\log\left(\frac{\ell_{\max}}{\beta}\right) - A\right). \tag{23}$$

Now we set $A = \log\left(\frac{1}{2\beta}\right)$ to obtain

$$\hat{\ell} \cdot \log\left(\frac{1}{2\beta}\right) \le \ell_A^* \cdot \left(\log\left(\frac{1}{2\beta}\right) + 8\log\left(\frac{\ell_{\max}}{\beta}\right)\right). \tag{24}$$

Substituting Equation (15) into Equation (21) gives

$$\underset{(\hat{\ell},\hat{\nu}) \leftarrow \mathcal{M}(D)}{\mathbb{P}}\left[\hat{\nu} \ge \max_{\ell \in [\ell_{\max}]} \widehat{\mathrm{DC}}(D;\ell) - \frac{2\ell + 2\ell_A^*}{\varepsilon}\log\left(\frac{1}{2\beta}\right) - \frac{8\ell + 16\ell_A^*}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right)\right] \ge 1 - 2\beta. \tag{25}$$

We simplify Equation (25) using $\log\left(\frac{1}{2\beta}\right) \le \log\left(\frac{\ell_{\max}}{\beta}\right)$ to obtain

$$\underset{(\hat{\ell},\hat{\nu}) \leftarrow \widehat{\mathcal{M}}(D)}{\mathbb{P}}\left[\hat{\nu} \ge \max_{\ell \in [\ell_{\max}]} \widehat{\mathrm{DC}}(D;\ell) - \frac{10\ell + 18\ell_A^*}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right)\right] \ge 1 - 2\beta. \tag{26}$$

Note that $\widehat{\mathrm{DC}}(D;\ell) \ge \frac{1}{2}\mathrm{DC}(D;\ell)$; hence,

$$\underset{(\hat{\ell},\hat{\nu}) \leftarrow \widehat{\mathcal{M}}(D)}{\mathbb{P}}\left[\hat{\nu} \ge \max_{\ell \in [\ell_{\max}]} \frac{1}{2}\mathrm{DC}(D;\ell) - \frac{10\ell + 18\ell_A^*}{\varepsilon}\log\left(\frac{\ell_{\max}}{\beta}\right)\right] \ge 1 - 2\beta.$$

Finally, note that $\ell_A^* \le \ell_*$; therefore, we proved Equation (7).

It only remains to verify that Algorithm 2 can be implemented in $O(|D|)$ time. We can implement S using a hash table to ensure that we can add an element or query membership of an element in constant time. (We can easily maintain a counter for the size of S.) We assume D is presented as a linked list of linked lists representing each u_i and furthermore that the linked lists u_i are sorted in lexicographic order. The outer loop proceeds through the linked list for $D = (u_1, \cdots, u_n)$. For each u_i, we simply pop elements from the linked list and check if they are in S until either we find $v \in u_i \setminus S$ (and add v to S) or u_i becomes empty (in which case we remove it from the linked list for D.) Since each iteration decrements $|D|$, the runtime is $O(|D|)$. $\square$