MIDE: Accuracy Aware Minimally Invasive Data Exploration For Decision Support

Sameera Ghayyur
University of California, Irvine
sghayyur@uci.edu

Dhrubajyoti Ghosh
University of California, Irvine
dhrubajg@uci.edu

Xi He
University of Waterloo
xi.he@uwaterloo.ca

Sharad Mehrotra
University of California, Irvine
sharad@ics.uci.edu

ABSTRACT

This paper studies privacy in the context of decision-support queries that classify objects as either true or false based on whether they satisfy the query. Mechanisms to ensure privacy may result in false positives and false negatives. In decision-support applications, often, false negatives have to remain bounded. Existing accuracy-aware privacy preserving techniques cannot directly be used to support such an accuracy requirement and their naive adaptations to support bounded accuracy of false negatives results in significant privacy loss depending upon distribution of data. This paper explores the concept of *minimally-invasive* data exploration for decision support that attempts to minimize privacy loss while supporting bounded guarantee on false negatives by adaptively adjusting privacy based on data distribution. Our experimental results show that the MIDE algorithms perform well and are robust over variations in data distributions.

PVLDB Reference Format:
Sameera Ghayyur, Dhrubajyoti Ghosh, Xi He, and Sharad Mehrotra. **MIDE:** Accuracy Aware **M**inimally **I**nvasive **D**ata **E**xploration For Decision Support. PVLDB, 15(11): 2653 - 2665, 2022.
doi:10.14778/3551793.3551821

1 INTRODUCTION

Decision-support (DS) applications [3, 11, 26] allow timely and informed decision-making and planning based on analyzing data, but such applications could face severe privacy challenges if the data analyzed contains personally identifiable information about individuals. For instance, a building management system may maintain the occupancy statistics (like in Figure 1) to detect violation of fire code, adherence to the CDC (Center For Disease Control) guideline in the context of COVID-19, or for better space utilization. If the location of interest has an aggregated occupancy that is higher than a threshold, an alarm is raised, but this aggregated statistics can leak sensitive information about users [10]. For example, prior work [17] has shown, with enough background knowledge, occupancy data can lead to inferences about the location of individuals, which, in turn, can leak sensitive information (*e.g.*, in an office building

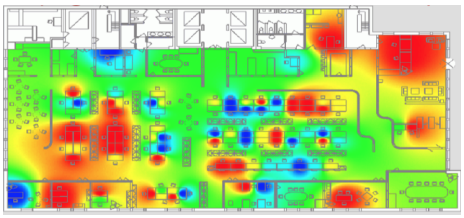


Figure 1: Occupancy Heatmap of a Building in UCI.

staff consistently leaving work early, smoking habits of individuals). As another example, consider assisted living situations where one of the primary challenges is fall prevention [28] of the elderly and the goal is to balance safety with privacy. We could monitor someone invasively using a camera, but such invasiveness is not necessary if the person is not a high fall risk. To make a decision about using invasive means of monitoring, wearables can be used to collect aggregated statistics *e.g.*, number of sudden accelerations in a week. Sudden accelerations exceeding a threshold could be interpreted to mean high fall risk and we can make a decision to monitor such an individual more invasively. The commonality in such DS applications is that the aggregated statistics are collected and compared to a preset threshold that classifies objects as either satisfying the predicate (*i.e.*, true), or as not satisfying the predicate (*i.e.*, false). Simply releasing the aggregated statistics, however, can lead to privacy violation of individuals , *i.e.*, reconstruction attack as shown in [4, 5, 7, 17].

Much of the prior work on privacy has been motivated by the need for data sharing while ensuring the privacy of sensitive data. Examples include privacy-preserving sharing of demographic data (*e.g.*, US Census), medical data to support research (*e.g.*, cancer registries), or collecting click-stream data for vulnerability analysis (*e.g.*, from browsers). Over the past decade, differential privacy [6] has emerged as one of the most popular privacy notions. It provides a formal mathematical guarantee that individual records are hidden even with the release of aggregate statistics and it is possible to bound the information leakage by a total privacy budget across multiple data releases. This has led to a wide range of adoption of differential privacy in a number of products at the US Census Bureau [8], Google [21], and Uber [12].

While differential privacy is suited for privacy-preserving sharing, its usefulness in the context of decision support (DS) applications is limited. DS tasks require guarantees on the output quality, especially, for false negatives that may result due to the addition of noise to aggregated statistics. Such false negatives may result in events of interest/anomalies not being detected. For instance,

Proceedings of the VLDB Endowment, Vol. 15, No. 11 ISSN 2150-8097.
doi:10.14778/3551793.3551821

in the elderly fall prevention example, a false negative may cause increased fall risk (from aggregated statistics of the number of accelerations) to go unnoticed preventing timely escalation and intervention. False positives are also not desirable, e.g., in the elderly fall prevention scenario, it may result in unnecessary escalation by using more invasive camera technology and wasted resources of video processing. Likewise, in the example of a fire code violation in a building, false positives on highly occupied spaces in the building may result in a heightened investigation of the region. While one would desire effective bounds on both false negatives and positives, in DS applications, increased false negatives are far more debilitating (compared to false positives) since they effectively defeat the very purpose of decision support. Thus, in DS applications, we desire to have bounded guarantees on false negatives without significantly increasing the number of false positives. [1]

Traditionally, DP-based approaches focus on providing formal privacy guarantees (in the form of a privacy parameter) while trying to maximize utility. These techniques do not offer guarantees on the quality of data outputted. Recent studies have addressed this challenge by designing accuracy aware DP techniques where the goal is to provide provable bounds on utility, e.g., [9, 20, 24]. Such approaches, however, are unsuitable for DS for several reasons: first, such approaches do not differentiate between false positives and false negatives, and offer a symmetric guarantee on both which makes them suboptimal in the DS context. Furthermore, the guarantee such approaches offer have a *region of uncertainty* around the threshold such that bounded guarantees (on either false positives or negatives) do not apply to data that falls in that region. This makes the techniques unsuitable for DS applications that require a tight guarantee on (at least) the false negatives.

In this work, we explore a utility-aware technique that provides (probabilistically) bounded guarantee on utility (in terms of asymmetric bounds on false negatives that are guaranteed to remain lower than a limited number) while minimizing privacy loss using differential privacy. The key intuition is to *modify* the DS query appropriately (before adding noise) so as to control the trade-off between false positives and false negatives and supports guaranteed utility in terms of false negatives. In particular, we generalize the query condition (e.g., replacing a query condition $X > \tau$ by $X > \tau'$, where $\tau' < \tau$) to admit a larger number of false positives but reduce the probability of data being wrongly classified as a false negative.

While a scheme that offers a bounded guarantee on false negatives can be designed by weakening the query condition, a proper design leads to subtle complexities. As will become evident, the (probabilistic) guarantee on false negatives, the weakening of the query condition, and the amount of privacy loss (ϵ in differential privacy terms) are interrelated. In particular, the weaker we make the query condition (i.e., over-generalization), the lower the privacy loss (smaller ϵ), while maintaining a bound on the false negatives. However, the weaker the query condition, the more the number of false positives. Ideally, we would like to weaken the condition as much as possible, as long as it does not cause false positives to arbitrarily increase. This depends upon the data distribution. Imagine, for instance, that there is almost no data (or very little data) around the threshold specified in the query — such would be the case, for instance, for outlier queries. In such a case, weakening the query condition significantly would be desirable since that would allow us to reduce privacy loss without increasing false positives, while still ensuring the required bounds on false negatives.

In this paper, we explore the design space of solutions alluded to in the discussion above. We first explore a single-step approach that minimally weakens/generalizes the query condition to achieve the bounded guarantee. We then explore a multi-step approach, wherein we aggressively make a decision to significantly weaken the query condition, and then, based on the outcome (*i.e.*, possibility of too many false positives) progressively refine the condition at the cost of loss of privacy (*i.e.*, larger ϵ), while maintaining false negative bounds. Like prior multi-step approaches of Apex [9], our multi-step approach also offers Ex-Post Differential Privacy [19] where the final privacy budget spent is determined after the completion of algorithm. Finally, we explore a data dependent version[2] of the multi-step algorithm that exploits the knowledge of data distribution learnt in previous steps to minimize the privacy loss.

In our algorithms, different objects/entities can be processed (*i.e.*, tested for threshold satisfaction) at different levels of privacy (ϵ). In the initial steps, the objects are processed at smaller ϵ (*i.e.*, higher privacy), and as the algorithm proceeds, some of the objects may be processed more invasively at higher values of ϵ with the goal of reduce the overall privacy loss. We, thus, refer to our approach as *Minimally Invasive Data Exploration (MIDE).*

The idea of different entities having different privacy levels has been studied in several pieces of prior work e.g. Personalized Differential Privacy [13], One-sided Privacy [15]. However, these works do not explore or provide a metric for overall privacy loss.

In summary, our contributions in this paper are as follows:

- We introduce and formally define the problem of accuracy aware privacy-preserving decision support that has wide applicability in privacy preserving applications.
- We introduce Predicate-wise Differential Privacy (referred to as PWDP) which is suited for a data dependent approach to accuracy aware privacy-preserving analysis. We formally define the associated privacy metric for PWDP.
- We develop multiple efficient algorithms for the problem of accuracy aware privacy preserving decision support, including a multi-step algorithm and its data dependent variant.
- We show the applicability of our approach in a detailed study of several real-world scenarios.

The organization of this paper is as follows: Section 2 defines basic concepts of differential privacy relevant to this work. Section 3 defines the decision support queries, accuracy requirements of such queries, and our problem statement. This section also provides a new privacy definition of Predicate-wise Differential Privacy (PWDP) and defines a new privacy metric to measure the privacy loss. We use this to minimize privacy loss for our accuracy aware differentially private decision support algorithms in Section 4. Section 5 provides an algorithm to compute the new privacy loss metric. In section 6, we evaluate our algorithms using multiple real datasets.

[1] If we ignored false positives and only considered false negatives, a trivial algorithm would be to simply ignore the query condition and return all the objects. This will meet the bounded requirement of false negatives and will have zero false negatives. But that also defeats the purpose of decision support applications.

[2] Data dependent algorithms have been studied in the context of differential privacy setting where privacy is fixed and we need to optimize utility [18, 29]

Lastly, we discuss future directions in Section 8. The paper contains several theorems and lemmas, the proofs of which are available in the longer version of the paper [2].

2 BACKGROUND

Differential privacy [6] has emerged as a widely used privacy definition with provable privacy guarantees. An algorithm is said to follow differential privacy given an input dataset $D \in \mathcal{D}$, if output of the algorithm does not change significantly, when a single tuple is added or removed from D. It is formally defined as follows:

Definition 1 (Differential Privacy (DP)). *A randomized mechanism $M : \mathcal{D} \rightarrow \mathcal{O}$ satisfies ϵ-differential privacy, if*

$$P[M(D) \in O] \leq e^{\epsilon} P[M(D') \in O] \tag{1}$$

for any set of outputs $O \subseteq \mathcal{O}$, and any pair of neighboring databases D, D' where D and D' differ by only one tuple, i.e., $| D \backslash D' \cap D' \backslash D | = 1$.

In this definition, ϵ is the privacy budget that controls the amount of privacy loss where $\epsilon \geq 0$. A higher ϵ value implies weaker privacy, whereas a lower ϵ value implies stronger privacy.

A Bayesian interpretation DP [14] is to bound the posterior odds of an adversary with respect to prior odds on whether a tuple x is in D and takes value $t \in \mathcal{T}$, where $\mathcal{T}$ is the domain of the tuples. The adversary's prior odds for the tuple x is defined as $\frac{P[x=t \wedge x \in D]}{P[x \notin D]}$, where the numerator refers to the prior belief that x is in the database and takes value t and the denominator denotes the prior belief that x is not in the database. The posterior odds after observing an output o of the DP mechanism M, is expressed as $\frac{P[x=t \wedge x \in D | o]}{P[x \notin D | o]}$. As M satisfies ϵ-DP, we have the following guarantees, given non-zero prior beliefs for x and t,

$$| \ln(\frac{P[x = t \wedge x \in D | o]}{P[x \notin D | o]} / \frac{P[x = t \wedge x \in D]}{P[x \notin D]}) | \leq \epsilon \tag{2}$$

The Laplace mechanism is one of the commonly used DP mechanisms and it achieves ϵ-DP by adding noise drawn from a Laplace distribution that is proportional to the *sensitivity*.

Definition 2 (Sensitivity). *Given a function $g : \mathcal{D} \rightarrow \mathbb{R}^d$, the sensitivity of g is defined as the maximum L_1 distance between function outputs of any two neighboring databases D and D' that differ by only one tuple.*

$$\Delta g = \max_{\forall D, D'} \| g(D) - g(D') \|_1 \tag{3}$$

For instance, a counting query has a sensitivity of 1.

Theorem 1 (Laplace Mechanism (LM)). *Given a function $g : \mathcal{D} \rightarrow \mathbb{R}^d$, the Laplace Mechanism outputs g(D) + η, where η is a d-dimensional vector of independent random variables drawn from a Laplace distribution with the probability density function $p(x|\lambda) = \frac{1}{2\lambda} e^{-|x|/\lambda}$, where $\lambda = \Delta g / \epsilon$, and it satisfies ϵ-DP.*

Differential privacy has important properties [6, 18] to allow the composition of multiple DP mechanisms.

Theorem 2 (Sequential Composition). *Consider k algorithms $M_1, ..., M_k$ each satisfying ϵ_i-DP. The sequential execution of $M_1, ..., M_k$ satisfies $\sum_{i=1}^{k} \epsilon_i$-DP.*

Theorem 3 (Parallel Composition). *Consider k algorithms $M_1, ..., M_k$, each satisfying ϵ_i-DP. The dataset D is partitioned into k disjoint parts and each M_i is executed on the i_{th} partition. Then the parallel execution of $M_1, ..., M_k$ satisfies $max(\epsilon_i)$-DP.*

3 PRIVACY IN DECISION SUPPORT

Decision support applications such as violation detection of the fire code based on the occupancy statistics or fall prevention based on weekly movement statistics, can be supported by a class of *aggregate threshold queries*. Such a query checks whether the aggregated values computed on a subset of tuples pass the thresholds or not.

Formally, an aggregate threshold query, denoted by $Q^{\Lambda}_{\mathrm{g}(.)>C}$, consists of (*i*) an aggregate function g(.); (*ii*) a set of predicates $\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_k\}$; and (*iii*) a set of corresponding thresholds $C = \{c_1, c_2, ..., c_k\}$. Each predicate λ_i takes in a tuple and outputs *True* or *False* based on the value of the tuple. Let D_{λ_i} be the set of tuples in D that evaluate λ_i to be *True*. This query returns all the predicates that have an aggregate $g(D_{\lambda_i})$ greater than their respective threshold c_i, i.e.,

$$Q^{\Lambda}_{\mathrm{g}(.)>C}(D) = \{\lambda_i \in \Lambda \mid \mathrm{g}(D_{\lambda_i}) > c_i\} \tag{4}$$

For example, consider a location dataset inside a building with schema $Location\_Data(person, location, timestamp)$, a decision support application would like to learn which locations have more people than their maximum capacity. In this example, the predicate is conditioned on the location of a tuple, the aggregate is the number of people for a given location, and the threshold is the maximum capacity of that location. Another way to look at the problem is that the whole database could be viewed as points in a multi-dimensional space, and each predicate defines a subspace or a region. Given a set of such non-overlapping regions, the goal is to find the regions that contain points more than a certain threshold.

Answering such an aggregate threshold query with differential privacy guarantees has been considered in prior work [9, 20, 22], but these solutions may fail the accuracy requirements of a decision support application or demand an unnecessarily large privacy budget. Next, we will describe and formalize the accuracy requirement and privacy requirement for decision support queries.

Accuracy Requirement. Two types of errors can be made by a randomized mechanism that answers a decision support query defined in Eqn. (4): (*i*) *false positives*, predicates that have smaller aggregate values than the thresholds but appear in the output; (*ii*) *false negatives*, predicates that have bigger aggregate values than the thresholds but are not outputted. While both false negatives and positives impact the effectiveness of the decision support application, preventing false negatives is far more crucial than false positives. A false negative may prevent timely intervention (*e.g.*, in the context of fall detection, or room code violation) which might be the very purpose of the decision support application. False positives, on the other hand, may result in false alarms that might have negative consequences in terms of wasted resources and/or violation of privacy (*e.g.*, as in more invasive monitoring in the fall detection example mentioned earlier). While one would like to minimize both, bounding false negative is far more crucial in decision support compare to false positives.

We formalize this accuracy requirement as follows.

Definition 3 (Accuracy Requirement (β-False Negative Rate)). *We say a mechanism $M : \mathcal{D} \rightarrow O$ satisfies β-false negative rate for an aggregate threshold query $Q^{\Lambda}_{g(.)>C}$ if for any database $D \in \mathcal{D}$, we have*

$$\forall \lambda_i \in \Lambda, P[\lambda_i \notin M(D) | \lambda_i \in Q^{\Lambda}_{g(.)>C}(D)] \leq \beta \tag{5}$$

Prior DP mechanisms such as the Laplace mechanism (Theorem 1) add noise from zero-mean distribution to the aggregate and compare it with the threshold, which place equal weights on false positives and false negatives. This approach can fail to bound both errors together by setting the privacy budget too small (large noise); or have guarantees on both false positives and false negatives, but with a high privacy cost. This symmetrical guarantee will be illustrated in Section 4.1. To bound the false negative rate without incurring additional privacy cost, we design a class of mechanisms that generalizes the thresholds in the query. For example, for an aggregate threshold query where we are checking $X > c$ for an aggregate X, we generalize the query threshold to $X > c - \alpha$. This type of generalization allows us to achieve trade-off between false negatives and false positives that helps us achieve β-false negative rate with a minimal privacy cost. This generalization parameter α and the accuracy parameter β are translated to privacy cost ϵ. We will present these algorithms in Section 4.

Privacy Requirement. The privacy budget (ϵ) of a DP mechanism depends on the accuracy specification (e.g. β in Def. 3). Furthermore, if the DP mechanism is data-dependent, then the minimum privacy budget to achieve the accuracy requirement also varies among the data and depends on the output. This privacy loss is known as *ex-post DP* [19]. If running the DP mechanism on the disjoint part of the data (based on the predicates) in parallel, each part of the data may end up with different ex-post privacy loss. For example, to achieve the same β-false negative rate, a predicate with an aggregate value that is far from the threshold can tolerate a large generalization parameter α and result in a small privacy loss; while another predicate that is close to the threshold requires a big privacy budget. To capture this predicate-wise privacy loss for DP applications, we propose a new framework Predicate-wise Differential Privacy to generalize DP and ex-post DP. This framework allows the decision support application to attain the required level of utility while using higher privacy levels for some predicates and lower privacy levels for other predicates.

3.1 Predicate-wise Differential Privacy

Consider a set of mutually exclusive predicates $\{\lambda_1, \lambda_2, ..., \lambda_k\}$ that they can partition a dataset D into disjoint parts $\{D_{\lambda_1}, D_{\lambda_2}, ..., D_{\lambda_k}\}$. We define the new privacy as follows. In this new framework, there is a privacy parameter ϵ_i associated with each predicate λ_i.

Definition 4 (Predicate-wise Differential Privacy (PWDP)). *Given $\Theta = \{(\lambda_1, \epsilon_1), (\lambda_2, \epsilon_2), ..., (\lambda_k, \epsilon_k)\}$, a set of mutually exclusive predicates that partition the full domain of the database and their corresponding privacy budgets, we say a randomized mechanism M satisfies Θ-Predicate-wise DP if for all i, for any neighboring databases D and D' differing in a record that satisfies λ_i, denoted by $D \sim_i D'$ i.e., $|\ (D_{\lambda_i} \setminus D'_{\lambda_i}) \cap (D'_{\lambda_i} \setminus D_{\lambda_i})\ | = 1$ and $D_{\lambda_j} = D'_{\lambda_j}$ for all $j \neq i$, the following condition holds:*

$$Pr[\mathcal{M}(D) \in O] \leq e^{\epsilon_i} \times Pr[\mathcal{M}(D') \in O] \tag{6}$$

In this new definition, the neighboring databases still differ by a single record (adding/removing a record), but the output distribution ratio depends on the value of the record. For example, for a location dataset inside a building with schema $Location\_Data(person, location)$, if it consists of only two predicates $\lambda_1 = (location = room_1)$ and $\lambda_2 = (location = room_2)$. Adding or removing a tuple $(person_1, room_1)$ will only affect the aggregate for only one of the above predicates (i.e. λ_1) as predicates are mutually exclusive. If this record takes a value t that satisfies one of the predicates λ_i and hence fails other predicates, then output distribution ratio is bounded by e^{ϵ_i}. A simple approach to achieve a predicate-wise DP is to run an ϵ_i-DP mechanism on a data partition D_{λ_i}.

Theorem 4. *Given $\Theta = \{(\lambda_1, \epsilon_1), (\lambda_2, \epsilon_2), ..., (\lambda_k, \epsilon_k)\}$, a set of mutually exclusive predicates and their corresponding privacy budgets, running ϵ_i-DP mechanism M_i over D_{λ_i} in parallel for $i = 1, \ldots, k$, achieves Θ-predicate-wise DP.*

It is also easy to see that a Θ-predicate-wise DP mechanism satisfies ϵ-DP, where $\epsilon = \max_{\epsilon_i \in \Theta} \epsilon_i$ by parallel composition of DP.

Predicate-wise DP also has the following composition properties. If two mechanisms consider different sets of mutually exclusive predicates, then the composed guarantee will create a new set of mutually exclusive predicates to partition the dataset further. If a new partition has participated in only one mechanism, it takes the privacy budget of that mechanism, and if it has participated in both mechanisms, it takes the sum of the two privacy budgets.

Theorem 5. *Let M_1 and M_2 be predicate-wise DP mechanisms with $\Theta_1 = \{(\lambda_1, \epsilon_1), ..., (\lambda_{k_1}, \epsilon_{k_1})\}$, and $\Theta_2 = \{(\lambda'_1, \epsilon'_1), ..., (\lambda'_{k_2}, \epsilon'_{k_2})\}$, respectively. Let $M = f(M_1(D), M_2(D))$, then M is Θ-predicate-wise DP with the following predicates and their respective privacy budgets:*

$$\Theta = \{(\lambda_i \wedge \lambda'_j, \epsilon_i + \epsilon'_j) \mid \forall (\lambda_i, \epsilon_i) \in \Theta_1, (\lambda'_j, \epsilon'_j) \in \Theta_2, \lambda_i \wedge \lambda'_j \neq \emptyset\} \tag{7}$$

where $\lambda_i \wedge \lambda'_j \neq \emptyset$ denotes that the two predicates overlap. We exclude the conjunctions of non-overlapping predicate pairs. The resulted predicate set is mutually exclusive and partitions the full domain.

Last, we provide the ex-post version of predicate-wise DP, that generalizes the ex-post DP [19]. We will use it for our data dependent algorithms.

Definition 5 (Ex-Post Predicate-wise DP). *Let $\mathcal{E} : O \rightarrow \mathbb{R}^{|\Theta|}$ be a function on the output space of a Θ-predicate-wise DP mechanism $M : \mathcal{D} \rightarrow O$. We say M satisfies $\mathcal{E}(o)$-Ex-post predicate-wise DP if for all $o \in O$, and any neighboring database D and D' differing in a record that satisfy λ_i,*

$$\max_{D,D':D \sim_i D'} \ln \frac{P[M(D) = o]}{P[M(D') = o]} \leq \mathcal{E}_i(o), \tag{8}$$

where $\mathcal{E}_i(o)$ denotes the ith entry of $\mathcal{E}(o)$, the ex-post privacy cost for predicate λ_i.

Theorem 6. *A PWDP mechanism $\mathcal{M}$ with $\Theta = \{(\lambda_1, \epsilon_1), ..., (\lambda_k, \epsilon_k)\}$ satisfies ϵ-DP with $\epsilon = \max_i \epsilon_i$. A mechanism $\mathcal{M}$ with an ex-post PWDP loss $\mathcal{E}(o)$ has an $\epsilon(o)$-ex-post DP with $\epsilon(o) = \max_i \mathcal{E}_i(o)$.*

PWDP can be used to track privacy loss in a more fine-grained manner (even without knowing the exact mechanisms) and result in a lower privacy loss even in terms of DP loss . Consider a database that only consists of two predicates λ_1, λ_2 to partition the domain. Consider two mechanisms M_1 and M_2, where the PWDP cost for M_1 is $\epsilon_{M_1,\lambda_1} = 0.1$, $\epsilon_{M_1,\lambda_2} = 0.5$ and the cost for M_2 is $\epsilon_{M_2,\lambda_1} = 0.5$, $\epsilon_{M_2,\lambda_2} = 0.1$. Keeping track of the fine grained epsilon loss per predicate using PWDP results in ex-post DP loss of 0.6. However, if we used DP, M_1 has a privacy loss of 0.5, and M_2 has a privacy loss of 0.5, and hence, the overall ϵ DP loss would be 1 by sequential/-parallel composition. Hence, a fine-graiend tracking of privacy loss allows a tigheter privacy analysis, and more queries to be answered with the same DP loss.

PWDP and its ex-post privacy can also be interpreted as providing bounds on adversarial posterior odds ratio just like DP. After observing an output o of a PWDP mechanism M, the adversary can not successfully distinguish whether a tuple x is in D and takes a value t that satisfies λ_i, denoted by t_{λ_i}or the tuple x is not in D. Given adversary's prior odds ratio *i.e.*, $P[x \in D \wedge x = t_{\lambda_i}]/P[x \notin D]$, the bounds on adversary's posterior odds ratio *i.e.*, $P[x \in D \wedge x = t_{\lambda_i}|o]/P[x \notin D|o]$ is as follows:

$$|\ln(\frac{P[x \in D \wedge x = t_{\lambda_i}|o]}{P[x \notin D|o]} / \frac{P[x \in D \wedge x = t_{\lambda_i}]}{P[x \notin D]})| \leq \epsilon_i \qquad (9)$$

Similarly, the ratio is bounded by $\mathcal{E}_i(o)$ for ex-post privacy.

3.2 Min-Entropy based Privacy Metric

Traditionally, DP mechanisms quantify privacy loss using ϵ. However, in predicate-wise DP, entities have different ϵ values. Comparing scenarios of different sets of epsilon values is non-trivial. For example, consider $(\epsilon_1 = 0.1, \epsilon_2 = 0.5, \epsilon_3 = 1)$ v.s. $(\epsilon_1 = 0.2, \epsilon_2 = 0.4, \epsilon_3 = 1)$ for three predicates, it is not obvious which scenario has a lower overall privacy loss as both have the same maximum epsilon value (1.0) and the same averaged epsilon value (0.53).

This section introduces our privacy metric for predicate-wise DP using *entropy*. In information theory, entropy is a well known metric for measuring uncertainty of a random variable. Given a discrete random variable X with possible outcomes of $x_1, ..., x_k$, with occurrence probabilities of $P(x_1), ..., P(x_k)$, the entropy of X is defined as: $-\sum_{i=1}^{k} P(x_i) \log P(x_i)$. In the context of predicate-wise DP, the adversary is guessing which predicate from the given set $\{\lambda_1, .., \lambda_k\}$ a record $x \in D$ can satisfy based on the output of a predicate-wise DP mechanism o. We use $\hat{p}_i$ to denote the posterior belief that x takes t_{λ_i}, a value satisfies λ_i. This posterior is proportional to $\bar{p}_i = \sum_{t_{\lambda_i}} (\frac{P[x \in D \wedge x = t_{\lambda_i}|o]}{P[x \in D|o]})$ and hence $\hat{p}_i = \bar{p}_i / \sum_i \bar{p}_i$. Then, the entropy over $\{\hat{p}_1, \ldots, \hat{p}_k\}$ can measure how uncertain the adversary' belief about the value of x.

There is no direct information for the posterior beliefs, but based on the predicate-wise DP guarantee (Eqn. (9)), we can derive a lower and upper bound for each posterior belief $\hat{p}_i$.

Lemma 7. *Given a Θ-Predicate-wise DP mechanism M with output o, where $\Theta = \{(\lambda_1, \epsilon_1), (\lambda_2, \epsilon_2), ..., (\lambda_k, \epsilon_k)\}$, each adversarial posterior guess $\hat{p}_i \propto \sum_{t_{\lambda_i}} \frac{P[x \in D \wedge x = t_{\lambda_i}|o]}{P[x \in D|o]}$ is bounded:*

$$\frac{e^{-\epsilon_i}}{\sum_i e^{\epsilon_i}} \leq \hat{p}_i \leq \frac{e^{\epsilon_i}}{\sum_i e^{-\epsilon_i}}, \qquad (10)$$

when priors $p_i \propto \sum_{t_{\lambda_i}} \frac{P[x \in D \wedge x = t_\lambda]}{P[x \in D]}$ are the same for $i \in [1, k]$.

This lemma assumes that the priors are the same for all predicates, which is possible when the adversary does not know the person. We also present the extended lemma for general priors in the appendix of our full paper [2]. Under these bounds, the largest entropy can always be attained when setting $\hat{p}_i$ the same for all the predicates. Hence, we consider the least uncertainty (min-entropy) as the privacy metric for predicate-wise DP.

Definition 6. *[Min-Entropy of PWDP] The privacy metric (Min-Entropy) of a Θ-Predicate-wise DP with $\Theta = \{(\lambda_1, \epsilon_1), ..., (\lambda_k, \epsilon_k)\}$ is defined as follows:*

$$\gamma(\Theta) = \min \sum_{i=1}^{k} -\hat{p}_i \log \hat{p}_i \qquad (11)$$
$$s.t. \frac{e^{-\epsilon_i}}{\sum_i e^{\epsilon_i}} \leq \hat{p}_i \leq \frac{e^{\epsilon_i}}{\sum_i e^{-\epsilon_i}} \ \forall i \in [1, k], \ and \ \sum_i \hat{p}_i = 1$$

Our privacy metric measures the lower bound on entropy, *i.e.*, the least uncertainty in the adversarial guess as $\gamma(\Theta)$. A high value of $\gamma(\Theta)$ means lower privacy loss, as the least uncertainty in adversarial guess is higher. Whereas, a low $\gamma(\Theta)$ means a higher privacy loss. We use this metric to compare the privacy loss of different Θs with the same set of predicates Λ. More details about an algorithm to compute this min-entropy metric are provided in §5.

3.3 Problem Definition

Consider the accuracy and privacy requirements defined above for decision support applications, we formalize our Accuracy Aware **M**inimally **I**nvasive **D**ata **E**xploration problem (or MIDE in short) as follows. Given an aggregate threshold query $Q^{\Lambda}_{g(.)>C}$ on a dataset D, we want to develop a set of differentially private mechanisms that answer the query with β-false negative rate guarantee (Def. 3) and minimal privacy loss in terms of ex-post privacy loss (Def. 5) and min-entropy (Def. 6). Among these mechanisms, we want to choose the DP mechanism with the minimal privacy loss.

4 ALGORITHMS FOR MIDE

In the section, we propose three algorithms that solve the MIDE problem. Recall that a decision support query $Q^{\Lambda}_{g(.)>c}(D)$ consists of a set of predicates $\Lambda = \{\lambda_1, ..., \lambda_k\}$, an aggregate function $g(.)$ and a set of thresholds $C = \{c_1, c_2, ..., c_k\}$. In this paper, we consider that the predicates in Λ are mutually exclusive and the aggregate function $g(.)$ is a counting function with sensitivity of 1. Extensions to other predicates and aggregates are discussed in the end.

All algorithms aims to satisfy the accuracy requirement of decision support query *i.e.*, the bound on β false negative rate (Definition 3). Our first algorithm is based on the modification of a previous work in the literature: APEx [9]. The second algorithm uses the concept of Predicate-wise DP (as introduced in §3.1) by iteratively increasing the privacy budget ϵ for each predicate till it reaches its accuracy bound. The third algorithm is a data dependent method that increases the privacy budget adaptively for different predicates in each iteration based on the outcome of the previous iterations.

4.1 Threshold-shift Laplace Mechanism

The Laplace Mechanism (Definition 1) can be used directly to answer the decision support query of $Q^{\Lambda}_{g(.)>C}$ in a privacy preserving

Algorithm 1 Threshold Shift Laplace Mechanism.

1: **procedure** ThresholdShiftLM($Q^{\Lambda}_{g(.)>C}, D, \alpha, \beta, \epsilon_{max}$)
2: $\epsilon \leftarrow \frac{\ln(1/(2\beta))}{\alpha}$
3: **if** $\epsilon \leq \epsilon_{max}$ **then**
4: $O \leftarrow \{\lambda_i \in \Lambda \mid g(D_{\lambda_i}) + \eta_i > c_i - \alpha, \eta_i \sim Lap(0, 1/\epsilon)\}$
5: **return** O, ϵ
6: **end if**
7: **return** 'Query Denied'
8: **end procedure**

manner. However, a naive application of this mechanism for this query can result in a large number of false positives and false negatives. We will first illustrate this limitation below, and then introduce an improved application, named as *Threshold-shift Laplace mechanism*, that achieves the required β-false negative rate.

Naive Laplace Mechanism. This mechanism adds a noise η_i to the aggregated count for each predicate λ_i, *i.e.*, $g(D_{\lambda_i})$, where $\eta_i \sim Laplace(0, 1/\epsilon)$. All predicates with noisy aggregate counts that are greater than the query thresholds *i.e.*, $g(D_{\lambda_i}) + \eta_i > c_i$ are returned as the query result. This randomized mechanism makes two types of errors in the output: (i) *false positives* which are the predicates with true aggregate $g(D_{\lambda_i}) \leq c_i$ but noisy aggregate $g(D_{\lambda_i}) + \eta_i > c_i$, ; (ii) *false negatives* which have true aggregate $g(D_{\lambda_i}) > c_i$ but noisy aggregate $g(D_{\lambda_i}) + \eta_i \leq c_i$.

If setting the privacy budget for Laplace Mechanism like prior work APEx [9] by $\epsilon = \frac{\ln(1/(2\beta))}{\alpha}$, we can achieve the following accuracy guarantees: with a small probability β, a predicate λ_i with a true aggregate $g(D_{\lambda_i}) > c_i + \alpha$ will have a noisy aggregate smaller than c_i (false negative); a predicate λ_i with a true aggregate $g(D_{\lambda_i}) < c_i - \alpha$ will have a noisy aggregate bigger than c_i (false positive). These guarantees are illustrated in Figure 2(i). However, no accuracy are guaranteed (bounded false positive/negative rates) for the predicates with true aggregates falling into the region of $[c_i - \alpha, c_i + \alpha]$. If most of the predicates have aggregates falling in to this uncertain region, the naive Laplace mechanism would output many predicates falsely and fail the accuracy requirement of decision support queries. One approach is to increase the privacy budget to shrink this uncertain region and hence reduce both false positives and false negatives. However, the decision support applications place more importance on the false negatives. We propose the following mechanism to bound the false negatives without increasing the privacy cost.

Threshold Shift Laplace Mechanism. This mechanism aims to achieve a bounded false negative rate for all the predicates (Definition 3) unlike the previous naive mechanism. Instead of comparing the noisy aggregates with the initial threshold C in the query $Q^{\Lambda}_{g(.)>C}$, this mechanism compares each noisy aggregate $g(D_{\lambda_i}) + \eta_i$ with a shifted threshold $c_i - \alpha$, where α is a generalized parameter and noise η_i is based on a privacy budget $\epsilon = \frac{\ln(1/(2\beta))}{\alpha}$. This mechanism then returns all the predicates that have noisy aggregates greater than the shifted thresholds, i.e. $g(D_{\lambda_i}) + \eta_i > c_i - \alpha$.

Figure 2(ii) illustrates the guarantees of the new mechanism. Due to the generalization of the threshold from c to $c - \alpha$, the uncertain region with no accuracy guarantees shifts from $[c - \alpha, c + \alpha]$ to

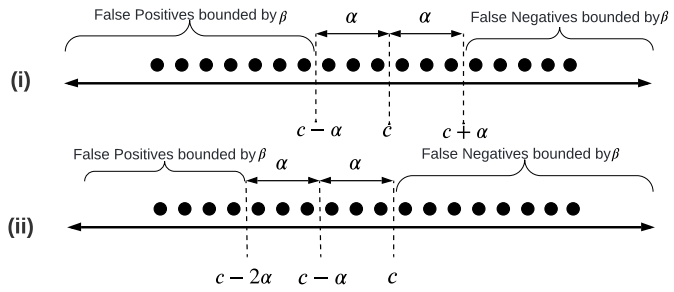


Figure 2: The figure shows accuracy guarantees of (i) Naive Laplace Mechanism: noisy aggregates are compared with threshold c (ii) Threshold Shift Laplace Mechanism: noisy aggregates are compared with shifted threshold $c - \alpha$. The dots represent aggregates on the predicates. By shifting the threshold to $c - \alpha$, (ii) achieves β-False Negative Rate (Definition 3) as compared to (i) where there is no guarantee on false negatives in the region $[c, c + \alpha]$

$[c - 2\alpha, c]$. This ensures that all the predicates with true aggregates greater than the original thresholds are in a guaranteed region, where they would have noisy aggregates smaller than the shifted thresholds and become false negatives with a small probability β.

This mechanism achieves β-false negative rate without increasing the privacy budget compared to the naive Laplace mechanism. Note that in this strategy, the false negative guarantee is independent of the choice of α, but such a guarantee comes at the cost of a potential increase of the false positives, which are the predicates with aggregates falling in the new uncertain region $[c-2\alpha, c]$. These predicates should not appear in the output as their true aggregate is smaller than the original thresholds, but their noisy aggregates are very likely greater than the shifted thresholds to output them. We name this region $[c - 2\alpha, c]$ as *α-uncertain region of false positives* for all mechanisms that use a threshold-shift approach. A larger generalization parameter α leads to a larger uncertain region, and can result in more false positives. We will use this generalized parameter α to limit the false positives.

Definition 7. *(Uncertainty Region) For each predicate $\lambda_i \in \Lambda$, the Uncertainty Region is based on the threshold $c_i \in C$ and the query generalization parameter α. It is defined the interval $[c_i - 2\alpha, c_i]$. If the predicate λ_i's aggregate value lies in this interval, the algorithm does not provide any bound on probability of λ_i to be in the output to the query as false positive.*

The Threshold-shift Laplace Mechanism is summarized in Algorithm 1. Given the β-false negative rate and α-uncertain region of false post as input, this algorithm first computes the minimal privacy budget to achieve these accuracy requirements, denoted by ϵ (line 2). It also takes the maximum privacy budget allowed for the query ϵ_{max} as input. If the budget is sufficient, then the algorithm proceeds with perturbing the aggregate for each predicate $g(D_{\lambda_i}) + \eta_i$ and returns the ones with noisy aggregates greater than the shifted thresholds $c_i - \alpha$ (line 4); otherwise, the query is denied (line 7). The guarantees of this algorithm are stated as follows.

Theorem 8. *Algorithm 1 satisfies ϵ_{max}-DP and β-false negative rate. If the query is not denied, its ex-post DP cost is $\epsilon = \frac{\ln(1/(2\beta))}{\alpha}$.*

4.2 Progressive Predicate-wise Laplace Mechanism

If we know that the aggregate value for a predicate λ_i is significantly smaller than its threshold, *i.e.*, $g(D_{\lambda_i}) << c_i$, then having a larger generalization α (which results in a smaller privacy loss) will still allow this predicate to stay out of the uncertain region of false positive, *i.e.*, $g(D_{\lambda_i}) < c_i - 2\alpha$.

Example 1. Consider two predicates λ_1, λ_2 with aggregates $g(D_{\lambda_1}) = 10$ and $g(D_{\lambda_2}) = 150$, which are smaller than their thresholds $c_1 = c_2 = 200$. To achieve $\beta = 0.01$-false negative rate using the Threshold Shift Laplace Mechanism, if generalizing the threshold from 200 to 120 by $\alpha = 80$ (which results in $\epsilon = \ln(1/2(0.01))/(80) = 0.049$), the first predicate with aggregate value 10 is still out of the uncertain region of false positives $[200 - 2 \cdot 80, 200]$ and it should be reported correctly with a high probability.However, the aggregate value of the second predicate falls into this $\alpha = 80$-uncertain region, and hence it requires a tighter generalization parameter, e.g. $\alpha' = 40$ to be in a region with guarantees, which leads to a larger privacy cost $\epsilon = \ln(1/2(0.01))/(40) = 0.098$. □

This observation motivates us to design an algorithm that provides different generalizations for the given predicates based on their aggregate values. Since the aggregate values $g(D_{\lambda_i})$ are unknown at first, we start each predicate with a large generalization parameter (and a small privacy budget), and incrementally tightens the generalization parameter (increases the privacy budget) till the predicate can be outputted or pruned with a high certainty. We name this algorithm *Progressive Predicate-wise Laplace Mechanism*, summarized in Algorithm 2.

Besides the same input as the Threshold Shift Laplace mechanism, Algorithm 2 takes in an initial privacy budget of ϵ_1 for the initial generalization and the number of iterations m. As each predicate can be tested at most m times, we aim β/m-false negative rate for each iteration to ensure that the overall false negative rate is bounded by β (Theorem 9). First, we estimate the total ϵ_m needed to satisfy the accuracy guarantee over m iterations. If the privacy budget is sufficient, $\epsilon_m < \epsilon_{max}$ (Line 2), we proceed the algorithm; otherwise, the query is denied.

The algorithm starts with ϵ_1 and its corresponding generalization α_1 in the first iteration (Lines 5-7). The algorithm increments ϵ_j in each iteration geometrically by a factor of $\omega = (\frac{\epsilon_m}{\epsilon_1})^{\frac{1}{m-1}}$ (Line 3), and the corresponding generalization parameter in the j-th iteration decreases by the same ratio. We consider geometric increments instead of arithmetic increments as smaller increments in the earlier iterations (*i.e.*, using smaller epsilon values) have a higher chance of achieving lower privacy loss. At the j-th iteration, the algorithm adds Laplace noise to the aggregate per predicate based on ϵ_j using Laplace mechanism or using PrivRelax [16]. PrivRelax generates noises for the next iteration j (noises based on ϵ_j) by drawing correlated noises based on the noise drawn in the previous iteration (noises generated using ϵ_{j-1}). This correlated noise ensures that the total privacy loss over the m iterations is bounded by ϵ_m.

Algorithm 2 Progressive Predicate-wise Laplace Mechanism

1: **procedure** ProgressivePWLM($Q^{\Lambda}_{g(.)>C}, D, \alpha, \beta, \epsilon_{max}, \epsilon_1, m$)
2: set final privacy cost $\epsilon_m \leftarrow \frac{\ln(1/(2\beta/m))}{\alpha}$
3: set $\epsilon_j \leftarrow \epsilon_1 \cdot \omega^{j-1}$ and $\alpha_j \leftarrow \frac{\ln(1/(2\beta/m))}{\epsilon_j}$ for $j = 1, \ldots, m$, where $\omega = (\frac{\epsilon_m}{\epsilon_1})^{1/(m-1)}$
4: **if** $\epsilon_m \leq \epsilon_{max}$ **then**
5: $[\eta_1, \ldots, \eta_{|\Lambda|}] \leftarrow Lap(1/\epsilon_1)^{|\Lambda|}$
6: $O_d \leftarrow \{\lambda_i \in \Lambda \mid g(D_{\lambda_i}) + \eta_i > c_i + \alpha_1\}$
7: $O_u \leftarrow \{\lambda_i \in \Lambda \mid g(D_{\lambda_i}) + \eta_i > c_i - \alpha_1 \wedge \lambda_i \notin O_d\}$
8: **for** $j = 2, \ldots, m$ **do**
9: **if** $O_u = \emptyset$ **then return** O_d, ϵ_{j-1}
10: **end if**
11: $[\eta_1, \ldots, \eta_{|\Lambda|}] =$ PrivRelax($\epsilon_{j-1}, \epsilon_j, [\eta_1, \ldots, \eta_{|\Lambda|}]$)
12: $O_d \leftarrow O_d \cup \{\lambda_i \in O_u \mid g(D_{\lambda_i}) + \eta_i > c_i + \alpha_j\}$
13: $O_u \leftarrow \{\lambda_i \in O_u \mid g(D_{\lambda_i}) + \eta_i > c_i - \alpha_j \wedge \lambda_i \notin O_d\}$
14: **end for**
15: **return** $O_u \cup O_d, \epsilon_m$
16: **end if**
17: **return** 'Query Denied'
18: **end procedure**

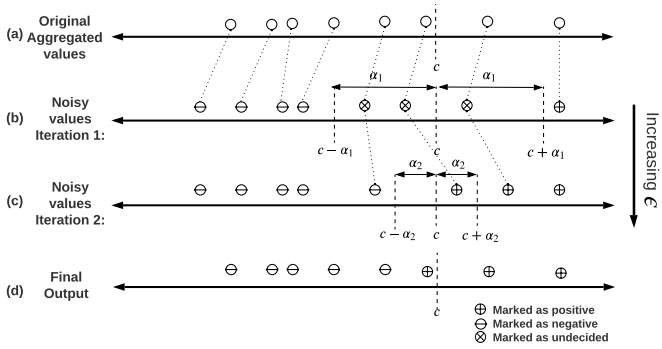


Figure 3: PPWLM with 2 iterations. (a) shows the original aggregated counts and the threshold c. (b) and (c) show the noisy aggregated values for each predicate for iteration 1 and iteration 2. In iteration 1, predicates with noisy aggregates $< c - \alpha_1$ are outputted as negatives, those with noisy aggregates $> c - \alpha_1$ are outputted as positives, the remaining are undecided and continue in iteration 2. Iteration 2 outputs all predicates with noisy aggregates $> c - \alpha_2$ as positives.

We categorize the predicates into three categories: (*i*) *decided*, denoted by O_d, which include predicates with noisy aggregates greater than the generalized thresholds and they are always outputted by the mechanism; (*ii*) *undecided*, denoted by O_u, which include the predicates with noisy aggregates in the range of $[c_i - \alpha_j, c_i + \alpha_j]$, and they are passed to the next iteration; and (*iii*) *eliminated*, which are the predicates with noisy aggregates lower than $c_i - \alpha_j$, and they are not considered in the next step or the output of the query. The union of O_d and O_u for each iteration is always a solution that achieves β-false negative rate like the Threshold Shift Laplace Mechanism, but by an iterative tightening of the generalization factor, the number of false positives are improved with a minimal

privacy loss. The algorithm terminates when the set O_u is empty *i.e.*, the algorithm has made decisions for all the predicates (Line 9). Otherwise, the algorithm terminates when it has spent the privacy budget of ϵ_m which satisfies the accuracy guarantees of α and β (Line 15). In this situation, the algorithm returns O_u as the answer of the query. The privacy loss in terms of ex-post DP or ex-post PWDP is dependent on the input data and releasing it breaks ϵ_{max}-DP. It is crucial that the ex-post (PW)DP loss is not released to the data analyst (adversary), as it will violate the ϵ_{max}-DP guarantees.

Theorem 9. *Algorithm 2 satisfies ϵ_{max}-DP and β-false negative rate. If the query is not denied, its ex-post DP cost is less than $\epsilon_m = \frac{\ln(1/(2\beta/m)}{\alpha}$.*

Figure 3 demonstrates the benefits of using this multiple step approach using $m = 2$. Figure 3(a) shows the true aggregated values of all predicates and the threshold c In the first iteration, the noisy aggregates (indicated by the position of the dots in Figure 3(b)) by spending ϵ_1 are compared against the corresponding generalized threshold $c - \alpha_1$. Four predicates marked negative have smaller noisy aggregates than $c - \alpha_1$ and are eliminated from the next iteration. Among the four predicates with noisy aggregates greater than $c - \alpha_1$, one of them has a noisy aggregate greater than $c + \alpha_1$ and hence it is directly outputted as a positive, while the other three continue to the next iteration. This iteration guarantees that there is a low probability $\beta/2$ for a predicate with true aggregate greater than c to be eliminated. In the second iteration, the newly perturbed aggregates with a larger privacy budget ϵ_2 (Figure 3(c)) are compared with a less generalized threshold $c - \alpha_2$. One additional predicate gets eliminated as its noisy aggregate is smaller than $c - \alpha_2$. The final output include 3 predicates. In this example, the final result does not contain any false negatives. Also, five predicates end up using ϵ_1 and three undecided predicates after iteration 1, end up using ϵ_2 privacy budgets. In some cases, the overall privacy loss can be smaller than the previous Threshold Shift Laplace mechanism, if we measure the privacy loss using ex-post Predicate-wise Differential Privacy and min-entropy $\gamma(\Theta)$ as described in §3.2.

4.3 Data Dependent Mechanism

The algorithm of previous section, (*i.e.*, Algorithm 2) used a fixed number of iterations and updated the privacy parameter and generalization parameter in a geometric manner. This section makes the case that this choice may not be optimal all the time. If the algorithm has knowledge about the data distribution, it can perform better in terms of privacy loss. Since we are using a multi-step algorithm, we can make use of the noisy aggregated values from the previous iteration to determine the number of iterations and the privacy/generalization parameters for the subsequent steps. We call this algorithm *Data Dependent Progressive Laplace Mechanism*, summarized in Algorithm 3. The privacy loss in terms of ex-post DP or ex-post PWDP is data dependent just like PPWLM so the ex-post (PW)DP loss is not released to the data analyst (adversary) in order to achieve ϵ_{max} DP guarantee.

Algorithm 3 first plans the privacy budgets (Lines 2 - 3), denoted by a vector B of m entries, in a way similar to Algorithm 2. In the first iteration, it still starts with ϵ_1 and stores the noisy aggregates G. Based on the noisy aggregates, the predicates are classified into three categories, decided positives O_d, undecided ones O_u, and

Figure 4: Possible options at k-th step of MinEnt algorithm. Option 1 distributes as much slack as possible to $\hat{p}_k$ (solid green line) and the rest to $\hat{p}_1, \ldots, \hat{p}_{k-1}$ (dotted green line). Option 2 distributes as much slack as possible to $\hat{p}_1, ..., \hat{p}_{k-1}$ and the rest to $\hat{p}_k$. Option 3 distributes slack to $\hat{p}_1, \ldots, \hat{p}_{k-1}$ and $\hat{p}_k$ instead of distributing as much as possible to either.

decided negatives ($\Lambda - O_d - O_u$). For all the predicates with a confident decision (*i.e.*, decided positives and decided negatives), their ex-post privacy cost stop at ϵ_1 and are saved in a vector E while the others in O_u are temporarily set to be the final cost ϵ_m (Line 9). In the next iteration, rather than using the planned privacy budget stored in B, we use the noisy aggregates G and the temporary ex-post privacy cost E to estimate the best privacy level that maximizes the min-entropy $\gamma(\Theta)$.

The estimation of the best privacy level for the next iteration is presented in Algorithm 4. It searches the privacy level ϵ_{next} for the next iteration in the remaining privacy levels in B and for each privacy level in B, it also further divides the intervals into m_f number of fine-grained steps (Line 4). The algorithm aims to find an ϵ_{next} that can lead to a predicate-wise privacy loss E' with a largest min-entropy; hence, the algorithm will be able to skip all the privacy levels before ϵ_{next} (Lines 5- 10). The algorithm removes the unused privacy levels from the budget plan B and updates the corresponding β for the next iteration (Line 12).

We cannot compute the exact predicate-wise privacy loss without running the algorithm. To estimate this privacy loss, the algorithm first uses the noisy aggregates G to compute how many of the undecided predicates from previous iteration O_u will still remain undecided if a privacy level of ϵ_{next} is used in the current iteration. For each predicate $\lambda_i \in O_u$, the algorithm estimates its probability of remaining undecided (*i.e.*, its new noisy aggregate $g(D_{\lambda_i}) + \eta_i'$ falls into the range of $[c_i - \alpha_j, c_i + \alpha_j]$) by using its noisy aggregate $G[i]$ which was perturbed by η_i at a privacy level ϵ_{j-1} from the previous iteration; and then sum them up as an expected number for the undecided predicates:

$$n_u = \sum_{\lambda_i \in O_u} P(g(D_{\lambda_i}) + \eta_i' \in [c_i - \alpha_j, c_i + \alpha_j]) \quad (12)$$

$$\approx \sum_{\lambda_i \in O_u} \int_{c_i-\alpha_j}^{c_i+\alpha_j} \int_{-\infty}^{\infty} \frac{\epsilon_{j-1}}{2} e^{-|x-G[i]|\epsilon_{j-1}} \times \frac{\epsilon_j}{2} e^{-|z-x|\epsilon_j} dxdz$$

Theorem 10. *Algorithm 3 satisfies ϵ_{max}-DP and β-false negative rate. If the query is not denied, its ex-post DP cost is $max(E)$.*

This data dependent algorithm comes at computation cost as we choose ϵ in each iteration based on min-entropy. In the worst case scenario, the cost of computing min-entropy can be exponential in terms of number of predicates; hence it may incur high computation overhead the when number of predicates are very high. We present an efficient algorithm to compute this cost next.

Algorithm 3 Data Dependent Progressive PWLP

1: **procedure** DPPWLM($Q^{\Lambda}_{g(.)>C}, D, \alpha, \beta, \epsilon_{max}, \epsilon_1, m, m_f$)
2: set final privacy cost $\epsilon_m \leftarrow \frac{\ln(1/(2\beta/m))}{\alpha}$
3: set $B[j] = \epsilon_1 \omega^{j-1}$ for $j \in [1, m]$, where $\omega = (\frac{\epsilon_m}{\epsilon_1})^{1/(m-1)}$
4: **if** $\epsilon_m \leq \epsilon_{max}$ **then**
5: $[\eta_1, \ldots, \eta_{|\Lambda|}] \leftarrow Lap(1/\epsilon_1)^{|\Lambda|}$
6: set $G[i] = g(D_{\lambda_i}) + \eta_i$ for $\lambda_i \in \Lambda$ and $\alpha_1 = \frac{\ln(1/(2\beta/m))}{\epsilon_1}$
7: $O_d \leftarrow \{\lambda_i \in \Lambda \mid G[i] > c_i + \alpha_1\}$
8: $O_u \leftarrow \{\lambda_i \in (\Lambda - O_d) \mid G[i] > c_i - \alpha_1\}$
9: Initialize predicate epsilon $E[i] = \epsilon_1$ if $\lambda_i \in (\Lambda - O_u)$; for the other predicates, it with final cost $E[i] = \epsilon_m$
10: initalize $j \leftarrow 1$
11: **while** $\epsilon_j \leq \epsilon_m$ and $O_u \neq \emptyset$ **do**
12: $j \leftarrow j + 1$
13: $\epsilon_j, \beta_j, B = \text{NextStepParams}(E, G, B, O_u, \beta, m, m_f)$
14: $[\eta_1, \ldots, \eta_{|\Lambda|}] = \text{PrivRelax}(\epsilon_{j-1}, \epsilon_j, [\eta_1, \ldots, \eta_{|\Lambda|}])$
15: set $G[i] = g(D_{\lambda_i}) + \eta_i$ for $\lambda_i \in O_u$, $\alpha_j = \frac{\ln(1/(2\beta_j))}{\epsilon_j}$
16: $O_d \leftarrow O_d \cup \{\lambda_i \in O_u \mid G[i] > c_i + \alpha_j\}$
17: $O'_u \leftarrow O_u, O_u \leftarrow \{\lambda_i \in (O_u - O_d) \mid G[i] > c_i - \alpha_j\}$
18: set predicate epsilon $E[i] = \epsilon_j$ if $\lambda_i \in (O'_u - O_u)$
19: **end while**
20: **return** $O_u \cup O_d, \epsilon_j$
21: **end if**
22: **return** 'Query Denied'
23: **end procedure**

Algorithm 4 Estimated Epsilon for next step in DPPWLM

1: **procedure** NextStepParams($E, G, B, O_u, \beta, m, m_f$)
2: initalize $ent_{max} = 0, \epsilon_{next} = B[0]$,
3: $r_{opt} = 1, \epsilon_{opt} = \epsilon_{next}$
4: **for** $r \in [1, \ldots, |B| - 1]$ and $s \in [1, \ldots, m_f]$ **do**
5: $\epsilon_{next} \leftarrow \epsilon_{next} + \frac{(B[r+1] - B[r])}{m_f}$
6: $E' \leftarrow E$ and choose $(|O_u| - n_u)$ number of predicates from O_u and set their $E'[\lambda_i] = \epsilon_{next}$
7: ent_{next}= MinEnt($b_{E'}$) ▷ $b_{E'}$ are bounds on $\hat{p}_i$ (Eq 10) based on E'.
8: **if** $ent_{max} \leq ent_{next}$ **then**
9: $ent_{max} \leftarrow ent_{next}, r_{opt} \leftarrow r, \epsilon_{opt} \leftarrow \epsilon_{next}$
10: **end if**
11: **end for**
12: **return** $(\epsilon_{opt}, \frac{\beta \cdot r_{opt}}{m}, B[r_{opt} + 1 :])$
13: **end procedure**

5 COMPUTING PRIVACY LOSS

We use an entropy based privacy metric for PWDP to compute the privacy loss of our multi-step algorithms (*i.e.*, PPWLM and DPPWLM). Furthermore, we use this metric to estimate the optimal ϵ values in each iteration to minimize the privacy loss in DPPWLM.

Our privacy metric for PWDP measures the lower bound on entropy, *i.e.*, the least uncertainty in the adversarial guess as follows: $\gamma(\Theta) = min(\sum_{i=1}^{k} -\hat{p}_i \log \hat{p}_i)$, subject to $\frac{e^{-\epsilon_i}}{\sum_i e^{\epsilon_i}} \leq \hat{p}_i \leq \frac{e^{\epsilon_i}}{\sum_i e^{-\epsilon_i}}$ and $\sum_i \hat{p}_i = 1$. This is a concave optimization problem with constraints. Finding the global minima with constraints for a concave function is computationally difficult since the function may have several local minimas [25]. However, finding the minima of the sum of entropy functions is a tractable problem, since the shape of entropy function is known and simple (*i.e.*, with only one maxima instead of multiple local maxima). We leverage this idea to develop a dynamic programming based algorithm that finds the global minima of the sum of entropy functions, *i.e.*, to compute $\gamma(\Theta)$.

Given Θ, *i.e.*, a set of k predicates with their epsilons, the algorithm first computes their corresponding lower bounds ($l_i = \frac{e^{-\epsilon_i}}{\sum_i e^{\epsilon_i}}$) and upper bounds ($u_i = \frac{e^{\epsilon_i}}{\sum_i e^{-\epsilon_i}}$) and then sort them based on their upper bounds in ascending order as an input to Algorithm 5. For simplicity, we assume that $u_1 \leq u_2 \cdots \leq u_k$ without introducing new indices.

If we start by allocating each $\hat{p}_i$ with its lower bound l_i, there is a remaining amount $s = (1 - \sum_{i=1}^{k} l_i)$ which has to be distributed to among $\hat{p}_i$s to ensure $\sum_i \hat{p}_i = 1$ and $\hat{p}_i \leq u_i$ while minimizing the entropy function. We call this remaining amount *slack*. The maximum slack that can be distributed to $\hat{p}_i$ is bounded by $\Delta_i = u_i - l_i$. We consider three options that cover all possible distributions of the slack s among the k predicates:

- **Option 1**. Distribute as much slack as possible to the $\hat{p}_k$ (the one with the largest upper bound).
- **Option 2**. Distribute as little slack as possible to the $\hat{p}_k$, and hence distribute as much slack as possible to $\hat{p}_1, \ldots, \hat{p}_{k-1}$.
- **Option 3**. Unlike the previous two options, here the slack is divided between $\hat{p}_k$ and the sub-problem of size $k-1$ *i.e.*, $\hat{p}_{k-1}$, ..., $\hat{p}_1$ without fully allocating to either of them.

These three options are illustrated in Figure 4. The figure represents the interval of $[l_i, u_i]$ from $i = 1, \ldots, k$. Note that a lower ϵ_i value will have a higher l_i and a lower u_i value. For option 1, if the slack $s > \Delta_k$, there is still remaining slack to be distributed among the $k - 1$ predicates. This gives a sub-problem of size $k - 1$, i.e. distributing the new slack $s' = (s - \Delta_k)$ among the first $(k - 1)$ predicates. For option 2, if the slack $s < \sum_{i=1}^{k} \Delta_i$, then the remaining slack will be added to $\hat{p}_k$; otherwise, we need to solve a sub-problem of size $k - 1$, i.e., distributing the full slack s among the first $(k - 1)$ predicates. We don't need to solve additional sub-problem. For option 3, we can show that it always results in a poorer solution than the solution coming from option 1 or option 2.

Theorem 11. *Given a set of intervals of posterior probabilities $\{(l_i, u_i) \mid i = 1, 2, ...k\}$ and a slack s to be distributed among the intervals, the option 3 always performs worse than either the strategies of option 1 or option 2 in terms of minimizing entropy.*

Hence, Algorithm 5 considers only option 1 and option 2 and only option 1 requires solving a sub-problem with a smaller number of predicates. At the base case when $k = 1$, all the slack is allocated to this predicate (Line 2). When $k > 1$, we consider option 1 and option 2 described above. For option 1, the solution is stored in $p1$ (Lines 5-6) which requires solving a sub-problem for the first $(k-1)$ predicates with the remaining slack $s - min(\Delta_k, s)$. For option 2, the solution is stored in $p2$ (Lines 7-8) which requires solving a sub-problem for the first $(k - 1)$ predicates with the full slack s. The solution with higher entropy is returned.

Algorithm 5 Minimize Entropy

procedure MinEnt($\{(l_i, u_i) \forall i \in \{1, 2, ..., k\}\}, s$) ▷ sorted by u_i in ascending order. Initially, $s = (1 - \sum_{i=1}^{k} l_i)$ is a slack variable.
  if $k = 1$ **then return** $[l_1 + s]$
  end if
  $\Delta_i = u_i - l_i, \forall i \in \{1, 2, ..., k\}$
  $p1[k] = l_k + min(\Delta_k, s)$
  $p1[1 : k-1]$ =MinEnt($[l_i, u_i] \forall i \in \{1, ..., k-1\}, s - min(\Delta_k, s)$)
  $p2[1 : k-1]$ =MinEnt($[l_i, u_i] \forall i \in \{1, ..., k-1\}, min(\sum_{i=1}^{k-1} \Delta_i, s)$)
  $p2[k] = l_k + s - min(\sum_{i=1}^{k-1} \Delta_i, s)$
  if CalEntropy($p1$) < CalEntropy($p2$) **then return** $p1$
  else return $p2$
  end if
end procedure
procedure CalEntropy(p)
  return $-\sum_{i=1}^{|p|} p[i] \log(p[i])$
end procedure

Theorem 12. *Algorithm 5 outputs the optimal solution to the min-entropy problem $\gamma(\Theta)$.*

6 EXPERIMENTS

This section evaluates our algorithms (Algorithms 1,2, and 3) for MIDE using various queries taken from real life scenarios and over real datasets. This is to show that all the algorithms effectively achieve their accuracy guarantees in terms of bounded false negative rate; among them, the data dependent mechanism (Algorithm 3) obtains the lowest minimal privacy cost over most of the queries.

6.1 Setup

Datasets & Queries. We used two real-world datasets and designed queries for the evaluation as described below.

UCI Dataset. This dataset contains the occupancy data of 24 different buildings of University of California, Irvine campus collected in 2018 October [23]. The data consists of 3 million records where attributes are userID, location, time. The DS queries find out the anomalous incidents (*e.g.*, violation of fire safety norm setup by the California fire department), *i.e.*, buildings with occupancy (number of individuals) that was higher than their capacity. We run 2 queries: $Q1$ on a weekday (Oct 09) and $Q2$ on a weekend (Oct 13) that has different data distributions. Both queries check every hour between 7 a.m. to 10 p.m. if a building's occupancy is exceeding the threshold. Total number of predicates for both $Q1$ and $Q2$ are 15(number of hours)×24(number of buildings)= 420. $Q1$ and $Q2$ are also coupled with three levels of thresholds (high, medium, low), set as $1, 0.8$, and 0.6 times of the building capacities.

NYTaxi Dataset. This dataset records taxi trips in New York City in 2020 [1], consisting of 15.7 million records with 18 attributes, *e.g.*, pick-up and drop-off locations and their timestamps. We group the pickup locations into 34 different regions and run queries to find out the regions and timestamps that had anomalous pickup counts. We run two queries: $Q3$ is run on March (1-14) (before the lockdown); and $Q2$ is run on March (15-30) (after the lockdown). Both queries check for each day in the corresponding time range if a regions's pickup count was higher than the threshold for all 34 regions. Total number of predicates for $Q3$ are 34 (regions)×14(days)= 476 and for $Q4$ are 34 (regions)×16(days)= 544. For each predicate, we use the maximum number of pickups from Jan and Feb times a multiplicative factors of 1, 0.8, 0.6 as the high, medium, low thresholds.

We display the distributions of the absolute distance of the aggregates in each query from their corresponding thresholds in Figure 5. We use uniform priors for these datasets to compute min-entropy.

Algorithms & Parameters. We consider three MIDE algorithms: Threshold Shifted Laplace Mechanism (TSLM), Progressive Predicate-wise Laplace Mechanism (PPWLM), and Data Dependent PPWLM (DPPWL). The naive Laplace Mechanism (NLM) is evaluated at the same privacy cost as TSLM as a baseline for accuracy.

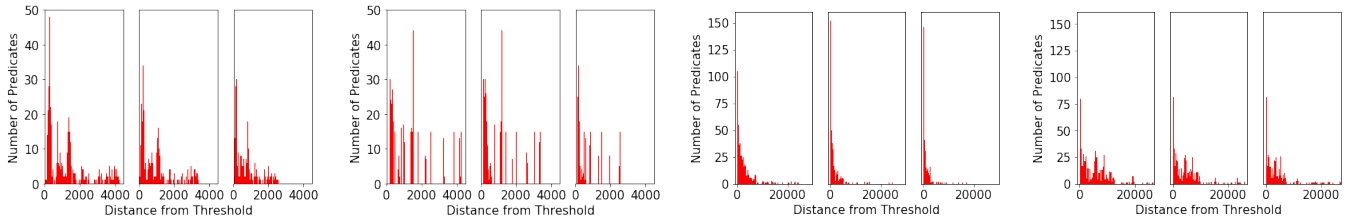
Our accuracy requirements is defined in terms of two parameters: β-false negative rate and α-uncertain region of false positives. We consider values for $\beta \in \{0.01, 0.02, \ldots, 0.1\}$ and $\alpha \in \{1, 10, 20, \ldots, 200\}$. The default values are $\beta = 0.05$ and $\alpha = 1$. For algorithms with multiple iterations including PPWLM and DPPWLM, we set the starting epsilon ϵ_1 be 0.00001, the total number of iterations to be $m = 4$, the maximum value without exceeding $\epsilon_{max} = 4$ at the default choice for α and β. For DPPWLM, we set the fine grained steps $m_f = 3$. We run each algorithm 100 times and report their averaged privacy or utility metrics.

6.2 Experimental Results

Privacy Results. We compare the algorithms based on two privacy metrics: ex-post DP, denoted by ϵ^*, and min-entropy for predicate-wise DP, $\gamma(\Theta)$. For TSLM, all predicates end with the same epsilon values, and hence the same lower and upper bounds for the posteriors to compute the min-entropy (Definition 6) using Algorithm 5. The privacy results for 4 queries ($Q1$-$Q4$) with their corresponding threshold levels (denoted by H,M,L) are presented in Figure 6 when setting the accuracy parameters $\beta = 0.05$ and $\alpha = 1$.

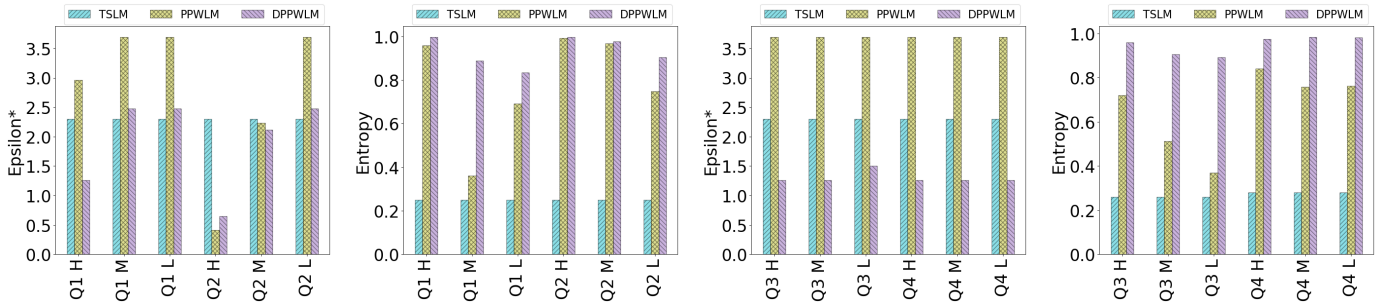
DPPWLM achieves a privacy cost that is near to the lowest or the lowest for all the queries. As it uses a multi-step approach, it allows earlier stop and hence a smaller ex-post DP cost than a single-step method TSLM for Q1H,Q2H/M, Q3H/M/L,Q4H/M/L, as shown in Figures 6a and 6c. DPPWLM does not always have an earlier stop, which depends on data distribution. For Q1M/L and Q2L, the distances of the counts from the thresholds shown in Figures 5a and 5b are relative small for most of the predicates, i.e., the counts are closer to thresholds. For such a case, all predicates need to consume a high privacy budget to be accurately decided and incur a slightly higher ex-post privacy than TSLM due to the division of the β among multiple steps. However, it is better than the other multi-step approach PPWLM, because DPPWLM uses learned data distribution to determines the number of iterations and hence budget allocation adaptively. Furthermore, as DPPWLM optimizes min-entropy, we observe that it achieves the highest min-entropy for all the queries as shown in Figures 6b and 6d.

Accuracy Results. For each run of the algorithm, we measured the number of false negatives n_{fn} and the number of false positives n_{fp}. Then we report the averaged false negative rate (FNR) as n_{fn}/n_p and the averaged false positive rate (FPR) as n_{fp}/n_n over multiple runs, where n_p and n_n are the number of positives and the number of negatives respectively. The results are presented in Figure 7 when $\beta = 0.05$ and $\alpha = 1$.



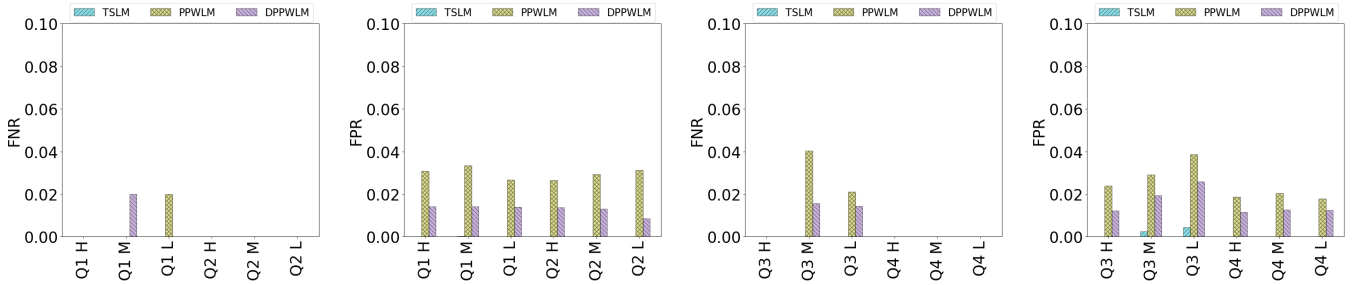
(a) Q1H/M/L (b) Q2H/M/L (c) Q3H/M/L (d) Q4H/M/L

Figure 5: This figure shows the distribution of the distances from the thresholds for all aggregates for $Q1, Q2, Q3, Q4$ with thresholds = High (H), Medium (M) and Low (L).



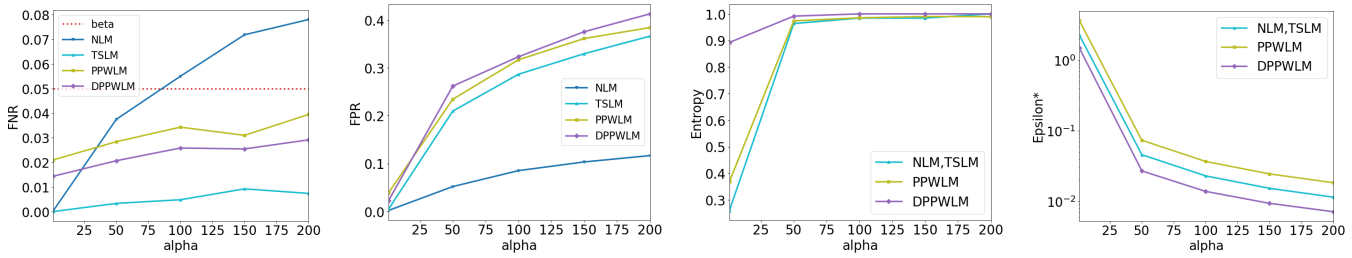
(a) UCI data ($Q1, Q2$) Epsilon (b) UCI data ($Q1, Q2$) Entropy (c) NYTaxi data ($Q3, Q4$) Epsilon (d) NYTaxi data ($Q3, Q4$) Entropy

Figure 6: Privacy loss in terms of ϵ^*(Ex-Post DP) and Min-Entropy $\gamma(\Theta)$ for Q1, Q2, Q3, Q4 with threshold = High (H),Medium (M), Low (L) at $\beta = 0.05$, $\alpha = 1$



(a) UCI data ($Q1, Q2$) FNR (b) UCI data ($Q1, Q2$) FPR (c) NYTaxi data ($Q3, Q4$) FNR (d) NYTaxi data ($Q3, Q4$) FPR

Figure 7: Accuracy in terms of False Negative Rate (FNR) and False Positive Rate (FPR) for Q1, Q2, Q3, Q4 with threshold = High (H), Medium (M), Low (L) at $\beta = 0.05$, $\alpha = 1$.



(a) FNR (b) FPR (c) Min-Entropy $\gamma(\Theta)$ (d) Ex-Post Privacy Loss ϵ^*

Figure 8: Accuracy (FNR,FPR) and Privacy (ϵ^*,$\gamma(\Theta)$) for $Q3$ (NYTaxi data) with threshold = Low over varying α.

Figures 7a and 7c show that all the MIDE algorithms achieve *a bounded FNR* lower than $\beta = 0.05$, which is the key accuracy requirement of DS. Note that the multi-step approach DPPWLM can make different decisions in each step (e.g., epsilon values) depending on the randomness of the algorithm and the data distribution, so there is no guarantee that DPPWLM will always win PPWLM in terms of utility (e.g., DPPWLM has a lower FNR than PPWLM for all queries except Q1M), but both of them have a bounded FNR. The trade-off of FNR in terms of FPR is relatively low, less than 0.04, for all MIDE algorithms and queries shown in Figures 7b and 7d.

Accuracy-Privacy Tradeoffs. TSLM achieves a better utility (FPR and FNR, and FPR/FNR tradeoff) than multi-step algorithms, but at a privacy cost. Since DPPWLM performs better than PPWLM (in both privacy cost and utility), we focus on the tradeoff comparisons between DPPWLM and TSLM. The privacy goal of DPPWLM is to optimize min-entropy (a higher min-entropy is preferred). The utility goal is to achieve a bounded FNR and optimize FPR (a smaller FPR is preferred). We compare its min-entropy (Figures 6b/6d) and its FPR (Figures 7b/7d) with TSLM. On average, DPPWLM improves the min-entropy of TSLM from 0.25 to a value above 0.8 in Figures 6b/6d, while it sacrifices the FPR of TSLM from ~0 to a value at most 0.034 in Figures 7b/7d for all the queries.

Comparison with Naive Laplace Mechanism. We use $Q3$ with threshold = 'low' for the comparison between the naive laplace mechanism (NLM) and our algorithms in Figure 8 by changing the accuracy parameter α. As there is no guideline for setting the parameter of NLM to achieve β-FNR, we use the same privacy budget for NLM as TSLM. When α increases, the privacy budget becomes smaller. Figure 8a shows that NLM does not satisfy β-false negative guarantee as α increases while the other algorithms still have a bounded FNR. Figure 8b shows that the trade off in terms of false positives for false negatives is data dependent. If many true negatives lie close to thresholds (most of our datasets), then the trade-off cost is high. The NLM has the same ex-post privacy loss and min-entropy as our TSLM as both algorithm use the same privacy budget. The results for our privacy metric (Figure 8c, 8d) show that our DPPWLM has the lowest privacy loss across different values of α. Similar results are observed when changing β.

Varying Parameters for Multi-step Algorithms. We evaluated our multi-step algorithms with varying starting epsilon values $\epsilon_1 \in \{10^{-5}, 10^{-4}, \ldots, 10^{-1}\}$ and varying number of steps $m \in \{2, 4, \ldots, 12\}$. Due to space constraints, we leave the plots in the appendix of our full paper [2] and summarize the results here.

As ϵ_1 increases, PPWLM and DPPWLM have a larger privacy loss (both ex-post DP and min-entropy). If ϵ_1 is too small, all predicates may be undecided in the first step in both approaches. However, DPPWLM chooses appropriate epsilons in the subsequent steps to effectively classify the predicates. When changing ϵ_1, there are no significant differences in utility and fulfilling the required accuracy bounds. The utility improves slightly if DPPWLM ends with a relatively higher privacy loss due to the data distribution and choice of ϵ and β in the intermediate steps.

Our experiments show that increasing the number of steps by more than 4 can result in a higher ex-post DP loss for both PPWLM and DPPWLM as ϵ_m for the last step will exceed $\epsilon_{max} = 4$. On the other hand, choosing a smaller number of steps may not result in an optimal solution as a data dependent algorithm becomes limited in the optimal choice of epsilon. The DPPWLM does better in min-entropy than PPWLM with a larger m as DPPWLM optimizes the choice of ϵ and β to maximize the min-entropy. The utility satisfies the required bound and varies slightly depending on the data distribution and the choice of ϵ and β across multiple steps.

7 RELATED WORK

Accuracy-aware differentially private (DP) systems [9, 19, 20, 24] have been studied in the literature. These systems allow data analysts to specify their accuracy requirements for their queries/ applications while achieving bounded privacy loss. However, queries supported by these systems or their accuracy specifications do not directly match the need for decision support applications.

Fine-grained privacy specifications similar to PWDP have been considered previously at tuple level, like personalized DP[13] where each tuple has its own pre-set privacy budget; or at group level, like one-sided DP [15] that specifies a set of tuples are non-sensitive based on their values. PWDP generalizes one-sided DP (a case with only two groups) by tracking the privacy budget at group level partitioned by the predicates. Both personalized DP and one-sided DP do not have any accuracy-aware designed algorithms. Predicate-wise DP can also be treated as a development over the parallel composition property [22] of DP.

In the context of privacy-preserving for decision support using DP, Cuong et al. [27] considered similar aggregate threshold queries, but they focus on optimizing a fairness goal for resource allocation for all the groups. Hence, the algorithms do not apply to our queries, and they did not take the accuracy-first approach. Extended related work can be found in the appendix of our full paper [2].

8 CONCLUSION AND FUTURE WORK

In this paper, we presented minimally invasive data exploration for decision support applications. We formally defined the accuracy requirement and presented three different privacy preserving algorithms that aim to minimize privacy loss while providing accuracy guarantees. Our results show that our data-dependent algorithm is robust and minimizes privacy loss for different data distributions. We limit the scope of this paper to binary classifiers using aggregate threshold queries. In future work, we would like to consider more general classifiers for decision support as generalizing a classifier to trade-off between false positive/ false negative applies to other types of classifiers. Another future direction is to generalize minimally invasive architecture for a broader class of SQL queries (e.g., queries with overlapping predicates and aggregate functions like *median* with higher sensitivity). Advanced DP mechanisms such as hierarchical mechanism [18] and exponential mechanism [6] can be applied, but accounting for their privacy loss in the predicate-wise DP framework will be an interesting problem. Last, we would like to explore fairness in the context of Predicate-wise DP as entities end with different privacy loss depending on the data distribution.

ACKNOWLEDGMENTS

This material was partially funded by the research sponsored by DARPA under agreement number FA8750-16-2-0021, NSF Grants No. 1952247, 2133391, 2032525, 2008993 and NSERC through a Discovery Grant.

REFERENCES

[1] 2020. TLC Trip Record Data. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page. Accessed: 2021-12-31.

[2] 2022. MIDE: Accuracy Aware Minimally Invasive Data Exploration. https://www.ics.uci.edu/~sghayyur/papers/MIDE_VLDB_2022.pdf. Accessed: 2021-12-31.

[3] Parag Chatterjee, Leandro J. Cymberknop, and Ricardo L. Armentano. 2017. IoT-based decision support system for intelligent healthcare — applied to cardiovascular diseases. In *2017 7th International Conference on Communication Systems and Network Technologies (CSNT)*. 362–366. https://doi.org/10.1109/CSNT.2017.8418567

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

[5] Cynthia Dwork, Frank McSherry, and Kunal Talwar. 2007. The price of privacy and the limits of LP decoding. In *Proceedings of the thirty-ninth annual ACM Symposium on Theory of Computing*. 85–94.

[6] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Found. Trends Theor. Comput. Sci.* 9, 3–4 (Aug. 2014), 211–407. https://doi.org/10.1561/0400000042

[7] Cynthia Dwork and Sergey Yekhanin. 2008. New efficient attacks on statistical disclosure control mechanisms. In *Annual International Cryptology Conference*. Springer, 469–480.

[8] Úlfar Erlingsson et al. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *ACM SIGSAC*. https://doi.org/10.1145/2660267.2660348

[9] Chang Ge et al. 2019. APEx: Accuracy-Aware Differentially Private Data Exploration *(SIGMOD)*.

[10] Sameera Ghayyur et al. 2018. IoT-Detective: Analyzing IoT Data Under Differential Privacy *(SIGMOD '18)*. ACM, New York, NY, USA, 1725–1728. https://doi.org/10.1145/3183713.3193571

[11] Nguyen Thi Ngoc Hien and Peter Haddawy. 2007. A decision support system for evaluating international student applications. In *2007 37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports*. F2A–1–F2A–6. https://doi.org/10.1109/FIE.2007.4417958

[12] Noah Johnson et al. 2018. Towards Practical Differential Privacy for SQL Queries. *Proc. VLDB Endow.* 11, 5 (Jan. 2018), 526–539. https://doi.org/10.1145/3187009.3177733

[13] Z. Jorgensen, T. Yu, and G. Cormode. 2015. Conservative or liberal? Personalized differential privacy. In *2015 IEEE 31st International Conference on Data Engineering*. 1023–1034.

[14] Daniel Kifer and Ashwin Machanavajjhala. 2014. Pufferfish: A Framework for Mathematical Privacy Definitions. *ACM Trans. Database Syst.* 39, 1, Article 3 (Jan. 2014), 36 pages. https://doi.org/10.1145/2514689

[15] I. Kotsogiannis, S. Doudalis, S. Haney, A. Machanavajjhala, and S. Mehrotra. 2020. One-sided Differential Privacy. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. 493–504.

[16] Fragkiskos Koufogiannis et al. 2015. Gradual Release of Sensitive Data under Differential Privacy. *CoRR* abs/1504.00429 (2015). arXiv:1504.00429 http://arxiv.org/abs/1504.00429

[17] Phillip Lee, Eun-Jeong Shin, Valerie Guralnik, Sharad Mehrotra, Nalini Venkatasubramanian, and Kevin T. Smith. 2019. Exploring Privacy Breaches and Mitigation Strategies of Occupancy Sensors in Smart Buildings. In *Proceedings of the 1st ACM International Workshop on Technology Enablers and Innovative Applications for Smart Cities and Communities* (New York, NY, USA) *(TESCA'19)*. Association for Computing Machinery, New York, NY, USA, 18–21. https://doi.org/10.1145/3364544.3364827

[18] Chao Li, Michael Hay, Gerome Miklau, and Yue Wang. 2014. A data-and workload-aware algorithm for range queries under differential privacy. *Proceedings of the VLDB Endowment* 7, 5 (2014), 341–352.

[19] Katrina Ligett, Seth Neel, Aaron Roth, Bo Waggoner, and Zhiwei Wu. 2017. Accuracy First: Selecting a Differential Privacy Level for Accuracy-Constrained ERM. *Journal of Privacy and Confidentiality* 9 (05 2017). https://doi.org/10.29012/jpc.682

[20] E. Lobo-Vesga, A. Russo, and M. Gaboardi. 2020. A Programming Framework for Differential Privacy with Accuracy Concentration Bounds. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 411–428. https://doi.org/10.1109/SP40000.2020.00086

[21] A. Machanavajjhala et al. 2008. Privacy: Theory meets Practice on the Map. In *2008 IEEE 24th International Conference on Data Engineering*. 277–286. https://doi.org/10.1109/ICDE.2008.4497436

[22] Frank D. McSherry. 2009. Privacy Integrated Queries: An Extensible Platform for Privacy-Preserving Data Analysis *(SIGMOD '09)*. Association for Computing Machinery, New York, NY, USA, 19–30. https://doi.org/10.1145/1559845.1559850

[23] S. Mehrotra et al. 2016. TIPPERS: A privacy cognizant IoT environment. In *2016 IEEE PerCom Workshops*.

[24] Prashanth Mohan, Abhradeep Thakurta, Elaine Shi, Dawn Song, and David Culler. 2012. GUPT: privacy preserving data analysis made easy. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. 349–360.

[25] Panos M Pardalos and J Ben Rosen. 1986. Methods for global concave minimization: A bibliographic survey. *Siam Review* 28, 3 (1986), 367–379.

[26] Rakesh Shirsath, Neha Khadke, Divya More, Pooja Patil, and Harshali Patil. 2017. Agriculture decision support system using data mining. In *2017 International Conference on Intelligent Computing and Control (I2C2)*. 1–5. https://doi.org/10.1109/I2C2.2017.8321888

[27] Cuong Tran, Ferdinando Fioretto, Pascal Van Hentenryck, and Zhiyan Yao. 2021. Decision Making with Differential Privacy under a Fairness Lens. https://doi.org/10.24963/ijcai.2021/78

[28] Andrea Ungar, Martina Rafanelli, Iacopo Iacomelli, Maria Angela Brunetti, Alice Ceccofiglio, Francesca Tesi, and Niccolò Marchionni. 2013. Fall prevention in the elderly. *Clinical Cases in mineral and bone metabolism* 10, 2 (2013), 91.

[29] Jun Zhang, Xiaokui Xiao, and Xing Xie. 2016. PrivTree: A Differentially Private Algorithm for Hierarchical Decompositions. In *Proceedings of the 2016 International Conference on Management of Data* (San Francisco, California, USA) *(SIGMOD '16)*. Association for Computing Machinery, New York, NY, USA, 155–170. https://doi.org/10.1145/2882903.2882928