Differentially Private Distributed Data Summarization under Covariate Shift *

Kanthi K. Sarpatwar [1]
IBM Research
sarpatwa@us.ibm.com

Karthikeyan Shanmugam [1]
IBM Research AI
karthikeyan.shanmugam2@ibm.com

Venkata Sitaramagiridharganesh Ganapavarapu
IBM Research
giridhar.ganapavarapu@ibm.com

Ashish Jagmohan
IBM Research
ashishja@us.ibm.com

Roman Vaculin
IBM Research
vaculin@us.ibm.com

Abstract

We envision Artificial Intelligence marketplaces to be platforms where consumers, with very less data for a target task, can obtain a relevant model by accessing many private data sources with vast number of data samples. One of the key challenges is to construct a training dataset that matches a target task without compromising on privacy of the data sources. To this end, we consider the following distributed data summarizataion problem. Given K private source datasets denoted by $[D_i]_{i\in[K]}$ and a small target validation set D_v, which may involve a considerable covariate shift with respect to the sources, compute a summary dataset $D_s \subseteq \bigcup_{i\in[K]} D_i$ such that its statistical distance from the validation dataset D_v is minimized. We use the popular Maximum Mean Discrepancy as the measure of statistical distance. The non-private problem has received considerable attention in prior art, for example in prototype selection (Kim et al., NIPS 2016). Our work is the first to obtain strong differential privacy guarantees while ensuring the quality guarantees of the non-private version. We study this problem in a Parsimonious Curator Privacy Model, where a trusted curator coordinates the summarization process while minimizing the amount of private information accessed. Our central result is a novel protocol that (a) ensures the curator accesses at most $O(K^{\frac{1}{3}}|D_s| + |D_v|)$ points (b) has formal privacy guarantees on the leakage of information between the data owners and (c) closely matches the best known non-private greedy algorithm. Our protocol uses two hash functions, one inspired by the Rahimi-Recht random features method and the second leverages state of the art differential privacy mechanisms. Further, we introduce a novel "noiseless" differentially private auctioning protocol for winner notification, which may be of independent interest. Apart from theoretical guarantees, we demonstrate the efficacy of our protocol using real-world datasets.

1 Introduction

Integrating new types of data to drive analytics based decision-making can contribute significant economic impact across a broad spectrum of industries including healthcare, banking, insurance,

*[1]Equal contribution by these authors.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

travel, and urban planning. This has led to the emergence of complex data ecosystems consisting of heterogeneous (and overlapping) data generators, aggregators, and analytics providers. In general, participants in these ecosystems are looking to monetize a class of assets that we term *AI assets*; such assets include raw and aggregated data, as well as models trained on such data. A recent Mckinsey global survey found, for example, that more than half of the respondents in sectors including basic materials and energy, financial services, and high tech stated that their companies had begun to monetize their data assets [Gottlieb & Khaled (2017)].

In light of the above, in this work we consider the basic setting of an AI Marketplace : a consumer arrives with a small dataset, referred to as a "validation" dataset, and wants to build a prediction model that performs well on this dataset. However, the model training process requires huge amount of data, that it must acquire from multiple private sources. Fundamentally, the AI Marketplace must address a *transfer learning problem*, where the distribution of data at different sources is considerably different from each other and even from the validation dataset. The Marketplace must facilitate transactions of data points from multiple sources towards the consumer's task by forming a training dataset that is close in some distance measure to the validation dataset. In the process, it must preserve data ownership and privacy as much as possible.

Consider the following scenario in the health care domain, as an example. Suppose the consumer is a newly established cancer hospital and the data sources are cancer institutions from different geographical locations across the globe. The goal of the new hospital is to construct ML models that, say, can predict early onset of some form of cancer. The quality of the model depends on the demography of its patients and therefore it is crucial to collect data that matches a small validation set that is representative of the demography. The individual sources clearly have widely different demographic data. The goal of an AI Marketplace is to enable private collection of a dataset sampled from these sources that matches the demography of the new institute. From the privacy perspective, there are two desirable properties: (a) The multiple data owners are typically "competitors" and therefore, individual data must be protected (for e.g. in a differentially private manner) from each other. (b) The platform (we use the term curator) must be "parsimonious" in handling data, i.e., it should access information on a "need to know" basis.

Motivated by the above, we consider the following problem. We consider K data owners with private datasets $D_1, \ldots, D_K$, and a data consumer who wishes to build a model for a specific task. The specific task is embodied by the consumer possessing a validation dataset D_v. The data consumer would like to procure a subset of data from each private dataset which is well-matched to its task. A parsimonious trusted curator does the collection of points. We call this the parsimonious curator model. Although trusted, we wish to minimize the number of points accessed by the curator to construct the the final summary. In turn, the K data owners would like to ensure that their data is private with respect to other data owners. The curator exchanges messages and data points with the data owners. We seek to make the exchanges by the curator to the data owners differentially private.

Our Contributions: We propose a novel protocol, based on an iterative hash-exchange mechanism, that enables the curator to construct a summary data set D_s from the K owner datasets. A central result of the paper shows that the proposed protocol simultaneously satisfies the following desired properties: (i) The constructed dataset D_s is well-matched to the validation dataset D_v, in terms of having a small Maximum Mean Discrepancy (MMD) (Gretton et al. (2008)); (ii) The protocol exchanges with any data owner i is (ϵ, δ)-differentially private with respect to the other owner datasets $\cup_{j \neq i} D_j$; and (iii) The parsimonious curator accesses at most $O(K^{\frac{1}{3}}|D_s| + |D_v|)$ data points. Qualitatively, we expect the protocol to produce data summaries that are useful for model building while maintaining differential privacy. We show through empirical evaluation that this is indeed the case; by examining generalization error on two example tasks, we show that the protocol pays only a small price for its differential privacy guarantees.

Prior Work: *Privacy Preserving Learning Algorithms:* There is a long line of work that considers private empirical risk minimization that seeks to optimize the trade-off between accuracy of a trained classifier and differential privacy guarantees with respect to the training set [Kasiviswanathan et al. (2011); Chaudhuri et al. (2011); Song et al. (2013); Kifer et al. (2012); Bassily et al. (2014); Shokri & Shmatikov (2015); Hamm et al. (2016); Wu et al. (2016); Abadi et al. (2016); Pathak et al. (2010); Thakurta (2013); Rubinstein et al. (2012); Talwar et al. (2015); Dwork et al. (2014a)]. One of the most notable in this line of work is the idea of adding noise to stochastic gradient iterations to preserve privacy [Song et al. (2013); Abadi et al. (2016); Shokri & Shmatikov (2015)]. We do not consider the

problem of learning a classifier directly. Our goal is to summarize diverse data sources in a distributed private manner to match a given validation set in a transfer learning setting. All the above works of privacy preserving learning algorithms can be applied after our summarization step. In Rubinstein et al. (2012); Chaudhuri et al. (2011), authors use noisy Rahimi-Recht Fourier features to release a representation of the support vectors for differentially private SVM classifier release. Our purpose of using Rahimi-Recht Fourier features is different and is used to expose the partial MMD objective at every round and in conjunction with novel private auctioning mechanisms.

Privately Aggregating Teacher Ensembles: Several works have considered the following setting: An ensemble of teacher classifiers, each trained on private data sources, noisily predict labels on an unlabelled public dataset that is further used to train a student model [Papernot et al. (2016)]. Again, this is different from our transfer learning setting where the various distributions are matched to a target task first handling covariate shift.

Another related line of work is differentially private submodular optimization [Mitrovic et al. (2017)]. While they consider a single private source, we handle multiple private data sources in optimizing a specific statistical distance (MMD). Our techniques leverage state of the art methods on privacy preserving mechanisms found in Dwork et al. (2014b); Hardt et al. (2012); Hardt & Rothblum (2010).

Domain Adaptation Methods: For the transfer learning problem, the existing domain adaptation methods [Ganin & Lempitsky (2014); Tzeng et al. (2014)] ensure the following: they learn a representation $\phi(\mathbf{x})$ such that $\phi(\cdot)$ of the source and the target are similar in distance (MMD metric has been used to regularize the distance penalty) and that classifying based on $\phi(\cdot)$ on the source have very high accuracy. However, most existing approaches used differentiable models like deep learning to achieve this - to learn $\phi(\cdot)$. In our methods, we first match the distributions in the ambient space by sub-selecting points and then train any suitable classifier. One advantage is that we can train any classifier after the moment matching step (Xgboost, Decision Tree, SVMs etc.). If one wants to make an existing domain adaptation algorithm private with respect to any pair of participants, one has to add noise to gradients computed at every step. The state of the art in differential privacy for deep learning [Abadi et al. (2016)] (in the non-transfer learning setting) adds Gaussian noise whose variance is linear in the number of iterations per step which significantly degrades the performance. In our method, we gain on this aspect as we add noise per point acquisition.

Federated Learning: We also note there is a distinction between our transfer learning setting from that of Federated learning McMahan et al. (2016). The validation set distribution is distinct from each of the individual data source distributions. There are significant covariate shifts between these. Federated learning would assume a training distribution which is obtained by sampling from different data sources uniformly at random or with a specific mixture distribution. In fact, in our experiments we contrast with training done on uniform samples which is a proxy for federated learning.

2 Problem Setting

The setting has K data owners with private datasets denoted by $D_1, D_2, \ldots D_K$. Here, $D_i \in \mathbb{R}^{m_i \times n}$ where m_i denotes the number of points and n denotes the their dimension. Further, there exists a "consumer" entity that wants to form a summary dataset (which can be used for downstream training goals) $D_s \subseteq \bigcup_i D_i$ and $|D_s| = p$. The quality of the summary set is measured by its *closeness* to a target validation dataset $D_v \in \mathbb{R}^{m \times n}$ which is private to the consumer. We measure the closeness of D_s to D_v using the MMD (Maximum Mean Discrepancy) statistical distance defined below.

Definition: The sample MMD distance for finite datasets $D \in \mathbb{R}^{m_1 \times n}$ and $D' \in \mathbb{R}^{m_2 \times n}$ is given by:

$$\mathrm{MMD}^2(D, D') = \frac{1}{m_1^2} \sum_{x, x' \in D} k(x, x') - \frac{2}{m_1 m_2} \sum_{x \in D, y \in D'} k(x, y) + \frac{1}{m_2^2} \sum_{y, y' \in D'} k(y, y') \qquad (1)$$

where $k(\cdot, \cdot)$ is a kernel function underlying an RKHS (Reproducing Kernel Hilbert Space) function space such that $k(x, y) = k(y, x)$ and $k(\cdot, \cdot)$ is positive definite.

Differential Privacy: We adopt the following definition of differential privacy [Dwork et al. (2006)] in our work. On a high level, it means that two datasets that differ in at most one point should not cause a differentially private algorithm to produce output that are very different statistically. Formally,

Definition: The output of a randomized algorithm $\mathcal{A}(D)$ is (ϵ, δ) differentially private with respect to the input dataset D if for any two neighboring datasets D, D' that differs in one data point,

$$P\left(\mathcal{A}(D) \in E\right) \leq e^{\epsilon} P\left(\mathcal{A}(D') \in E\right) + \delta. \tag{2}$$

for all events E that can be defined on the output space.

Parsimonious Curator Privacy Model: We assume that there exists a trusted curator, called *aggregator*, that collects the summary data points D_s. The participants holding data D_i wish to preserve the privacy of their individual data points. The model satisfies the following constraints:a) During the protocol run, the curator must not have access to more than $\rho(|D_s| + |D_v|)$ points. We refer to such a protocol as ρ-**parsimonious** protocol. The aggregator needs to collect points that closely match D_v in MMD distance. Therefore the aggregator at least sees $|D_s|$ points in this framework. This forms a natural $|D_s| + |D_v|$ lower bound on how many points the aggregator has to access. Therefore, we define a ρ-parsimonious aggregator who sees ρ times the minimum required. b) Communication to a non-trusted participant i is differentially private with respect to all other datasets i.e. $\cup_{j \neq i} D_i \cup D_v$. This setting can be viewed as an intermediate regime between the "centralized setting" and "localized setting" [Nissim & Stemmer (2017)] considered in the prior works. In other words, the source D_i knowing all but one point in the union of other datasets as side information must not know much (in the differential privacy sense) about the missing point given all the communication to it during the protocol (standard informed adversary model with respect to union of other datasets $\bigcup_{j \neq i} D_j$). Preservation of differential privacy across data sources constrains the aggregator to collect more points than necessary (i.e. $|D_s| + |D_v|$).

Main Problem: Is there a (ϵ, δ) differentially private protocol in the parsimonious curator model, that outputs a subset $D_s \subseteq \bigcup_i D_i : |D_s| = p$ that (approximately) minimizes $\mathbb{E}[\text{MMD}^2(D_s, D_v)]$?

Incentives: The aggregator needs to train a downstream task on a test distribution that is similar to D_v. To this end, $|D_s|$ points (much larger is size than D_v) are being collected for training. In fact, one could think of the aggregator paying for the points. Our protocol is approximately the best way to obtain such points. There is no incentive for the aggregator to cheat since it has to pay for the collected points. The data providers are happy to provide a set as long as they are compensated and other data sources do not know about their data (in a differential privacy sense.).

Every data source would be able to monetize their contribution in proportion to the value they provide to the summary. After the protocol ends, value of a data source's contribution could be deemed proportional to the sum of winning marginal bids from the source. Value attribution based on this would be a incentive for data holders to participate. We address the problem of value attribution to data sources in a companion paper (Sarpatwar et al. (2019)). We only focus on the privacy and parsimonious constraints.

Our Approach: We briefly summarize the greedy approach to solve the moment matching problem without privacy constraints. Our fundamental contribution is to make it differentially private in the parsimonious curator model.

Greedy Algorithm Without Privacy: Our objective is to form a summary D_s of size p by collecting points from all the data owners. We maximize the following normalized MMD objective [Kim et al. (2016)] as described below. For fixed validation set D_v such that $|D_v| = m$ and the summary set D_s, the objective $J(D_s)$ is as follows:

$$J(D_s) = \sum_{i,j \in D_v} \frac{k(y_i, y_j)}{m^2} - \text{MMD}^2(D_v, D_s) = \sum_{i \in D_v, j \in D_s} \frac{2k(y_i, x_j)}{m|D_s|} - \sum_{i,j \in D_s} \frac{k(x_i, x_j)}{|D_s|^2} \tag{3}$$

Note that our objective here is different from the one used in Kim et al. (2016), in that we do not have the property $S \subseteq V$. Submodularity of this function does not follow from their work directly. In Section E of appendix, we show that the function is submodular under some condition on the kernel function. This condition is satisfied if the distance between any two points is $\Omega(\sqrt{\log N})$ and when the RBF kernel $k(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|_2^2)$ is used with some constant $\gamma > 0$.

Theorem 1. *Let N be the total number of points in the system. Given a diagonally dominant kernel matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ satisfying $k_{i,i} = k^*$, for any $i \in [N]$ and $k_{i,j} \leq \frac{k^*}{N^3 + 3N^2 + N}$ for any $i \neq j$, then $J(S)$ is a non-negative, monotone and submodular function.*

It has been proven that the following iterative greedy approach yields a constant factor approximation guarantee, given that the objective is a non-negative monotone submodular [Nemhauser et al. (1978)] function. Iteratively, until the required summary size is achieved: (a) each participant computes its marginally best point y, i.e., that maximizes $J(D_s + y) - J(D_s)$ and (b) curator collects the marginally best points from various participants and adds the best among them to the summary.

Our Private Algorithm: The focus of the paper is to adapt this greedy approach with *privacy guarantees* in the parsimonious curator model. In our private protocol, the curator collects the data points in D_s in a greedy fashion as above. However, there is a key challenge on the privacy front:

Challenges: During the implementation of the greedy algorithm, the curator maintains a set of points $D_s = \{x_1, \ldots, x_k\}$. To calculate the marginal gain with respect to Equation (3), we observe that the curator needs to expose a function of the form $\sum \alpha_i k(x_i, \cdot)$ to every participant for some constants α_i (this will become clear later). However, sharing the points in the raw form would be a violation of privacy constraints at the participants. Further, over the course of multiple releases, any participant must not be able to acquire any information about previous data points of other participants. Therefore, the key issue is that the releases of the curator must be differentially private while enabling the computation of the (non-linear) marginal gain, over all the iterations of the protocol. Beyond enabling the computation of "best" points, privacy concerns also arise in the actual collection of data points. Indeed, even a private declaration of "winners" to data providers would result in the leakage of information on the quality of other data providers.

Our Solution: To solve these issues, we use two hash functions:

(a) $\mathfrak{h}_1(\cdot)$ based on the random Fourier features method of Rahimi-Recht to hash every data point at the curator. This hash function is common to all the entities (i.e., curator and the participants) and satisfies the property that $\mathfrak{h}_1(x)^T \mathfrak{h}_1(y) \approx k(x, y)$ w.h.p., which is useful to convert the non-linear kernel computation (Equation (3)) to a linear one. This enables approximate kernel computation by an entity external to the curator. Therefore, any entity can compute the marginal gain of a new point y by $\sum \alpha_i k(x_i, y) \approx \sum \alpha_i \mathfrak{h}_1(x_i)^T \mathfrak{h}_1(y)$. Thus, the curator needs to only share $\sum \alpha_i \mathfrak{h}_1(x_i)$.

(b) a second hash function $\mathfrak{h}_2(\cdot)$, whose randomness is private to the curator such that, $\mathfrak{h}_2(\sum \alpha_i \mathfrak{h}_1(x_i))^T \mathfrak{h}_1(y) \approx \sum \alpha_i k(x_i, y)$ and $\mathfrak{h}_2$ is differentially private with respect to $\mathfrak{h}_1(x_i)$. A specific participant can observe multiple releases of $\sum \alpha_i \mathfrak{h}_1(x_i)$ and potentially find out the last point that was added. Therefore, the releases of the sum vector $\sum \alpha_i \mathfrak{h}_1(x_i)$ needs to be protected.

Our $\mathfrak{h}_2(\cdot)$ is a novel adaptation of the well-known MWEM method [Hardt et al. (2012)]. The key technical challenge is to match the performance of the greedy algorithm while ensuring privacy properties of $\mathfrak{h}_2(\cdot)$ in order to protect data releases from the curator. Further, to address the privacy concerns in parsimonious data collection, we obtain a novel private auction mechanism that is $O(K^{\frac{1}{3}})$-parsimonious and (ϵ, δ)-differentially private, with no further loss in optimality. Aside from theory, we provide insights to make our protocol well-suited for practice and demonstrate its efficacy on real world datasets.

3 The Protocol

Our protocol uses two different hash functions that we refer to as $\mathfrak{h}_1(.)$ and $\mathfrak{h}_2(.)$. The hash function $\mathfrak{h}_1(.)$ is shared between the various data owners and the aggregator. The hash function $\mathfrak{h}_2(.)$ is used by the aggregator to hash the current summary dataset before being broadcast to various participating entities (owners). We now describe both the hash functions $\mathfrak{h}_1(\cdot), \mathfrak{h}_2(\cdot)$.

The Hash Function $\mathfrak{h}_1(\cdot)$: Our first hash function, which is shared and used by various data owners and the aggregator is based on a well known distance preserving hash function formulated by Rahimi & Recht (2008). Formally, the hash function is defined in Algorithm 1. The main purpose of this hash function is to ensure that $\mathfrak{h}_1(\mathbf{x})^T \mathfrak{h}_1(\mathbf{y}) \approx k(\|\mathbf{x} - \mathbf{y}\|)$. We assume an RBF kernel function throughout the paper which is given by $k(\Delta) = \exp(-\gamma \Delta^2)$. In Algorithm 1, $p(\omega)$ is the distribution defined by the Fourier transform of the kernel $k(\Delta)$, i.e. $p(\omega) = \frac{1}{2\pi} \int e^{-j\omega^T \Delta} k(\Delta) d\Delta$. Due to the RBF kernel, $p(\omega) = \mathcal{N}(0, 2\gamma \mathbf{I}_n)$. The randomness in the hash function is due to d random points drawn from this distribution as in Algorithm 1.

1: **Input:** Point $\mathbf{x} \in \mathbb{R}^n$, parameter γ, dimension parameter d
2: **Output:** $\mathfrak{h}_1(\mathbf{x})$
3: Draw $\{\omega_i\}_{i=1}^d$ i.i.d from the same distribution $p(\omega) = \mathcal{N}(0, 2\gamma \mathbf{I}_n)$ only **once** at the beginning of the protocol and reuse it over subsequent calls to $\mathfrak{h}_1(\cdot)$.
4: Draw samples $\{b_i\}_{i\in[d]}$ i.i.d uniformly from $[0, 2\pi]$ only **once** at the beginning of the protocol.
5: **return** $\mathfrak{h}_1(\mathbf{x}) = \sqrt{\frac{2}{d}} \left[\cos(\omega_1^T \mathbf{x} + b_1), \cos(\omega_2^T \mathbf{x} + b_2) + \ldots \cos(\omega_d^T \mathbf{x} + b_d)\right]^T$

Algorithm 1: Computing the hash function $\mathfrak{h}_1(\cdot)$.

The Hash Function $\mathfrak{h}_2(\cdot)$: Consider a dataset $D \in \mathbb{R}^{q\times d}$ consisting of vectors $\{\mathbf{v}_1, \mathbf{v}_2 \ldots \mathbf{v}_q\}$ such that $\mathbf{v}_i \in \mathbb{R}^{1\times d}$ and $-\sqrt{\frac{2}{d}} \leq v_{ij} \leq \sqrt{\frac{2}{d}}$, $1 \leq i \leq q$, $1 \leq j \leq d$. The hash function $\mathfrak{h}_2(D)$ approximately computes the vector sum $w(D) = \sum_i \mathbf{v}_i$ in a differentially private manner. Let $w(D, j) = \sum_i \mathbf{v}_{ij}$. We now provide the description of the $\mathfrak{h}_2(\cdot)$ in Algorithm 2. The algorithm has two components: (a) The algorithm first quantizes the q vectors in D to obtain D_Q such that the quantized coordinate values are from a grid S of points $S = \{-1, -1+\eta, -1+2\eta \ldots \ldots 1-\eta, 1\}$, for a parameter η (refer to Line 10 in Algorithm 2). (b) Then a random distribution P_{avg} over the space of all possible quantized vectors $S^{1\times d}$ is found such that the expected vector under this distribution is close to the sum of the quantized vectors in D_Q. Further, the releases are also differential private. This second part relies on the MWEM mechanism of Hardt et al. (2012).

Full Algorithmic Description of $\mathfrak{h}_2(\cdot)$: Let $\tilde{v}_1, \tilde{v}_2 \ldots \tilde{v}_q \in S^{1\times d}$ be the quantized vectors in D_Q and $w(D_Q, i) = \sum_{j=1}^q \tilde{v}_{ji}$. Now, we will define probability mass functions $P_t(\mathbf{s} \in S^{1\times d})$ for every time t over the finite set $S^{1\times d}$ whose cardinality is $|S|^d$. P_t will be dependent only on P_{t-1}. We will define the distribution iteratively over $t \leq T$ iterations. Define $w(P, i) = q(\sum_{s\in S} sP_i(s))$ with respect to a probability mass function P on $S^{1\times d}$ where $P_i(s)$ is the marginal pmf on the i-th coordinate. The way P_t is computed is given in Algorithm 2 (Steps 6-7).

1: **Input:** Dataset D, parameters ε, η and T
2: **Output:** $\mathfrak{h}_2(D, T, \varepsilon)$
3: Obtain $D_Q \leftarrow$ QUANTIZATION(D, η). Let P_0 be the uniform distribution over the set $S^{1\times d}$ where $S = \{-1, -1+\eta, -1+2\eta \ldots \ldots 1-\eta, 1\}$.
4: **for all** $t \leq [T]$ **do**
5: Sample a coordinate $i \in [d]$ with probability proportional to $\exp(\varepsilon\psi_i(D_Q))$ where the score function: $\psi_i(D_Q) = |w(P_{t-1}, i) - w(D_Q, i)|$. Let the sampled coordinate be $i(t)$
6: Let $\mu_{i(t)} \leftarrow w(D_Q, i(t)) + Lap(1/\varepsilon)$. Compute the distribution satisfying $P_t(\mathbf{s}) \propto P_{t-1}(\mathbf{s}) \exp\left[s_{i(t)}(\mu_{i(t)} - w(P_{t-1}, i(t)))/2q\right]$
7: **end for**
8: $P_{\text{avg}} = \frac{1}{T}\sum_{t\in[T]} P_t$.
9: **return** $\sqrt{\frac{2}{d}}\frac{1}{q}\left[w(P_{\text{avg}}, 1) \ldots w(P_{\text{avg}}, d)\right] = \mathfrak{h}_2(D, \varepsilon) = \mathfrak{h}_2(D_Q, \varepsilon)$

10: **procedure** QUANTIZATION(D, η)
11: Define $Q(x) = \left\{ \begin{array}{cc} -1+k\eta, & w.p. \frac{(k+1)\eta-1-x}{\eta} \\ -1+(k+1)\eta & w.p. \frac{x+1-k\epsilon}{\eta} \end{array} \right\}$ where $k = \lfloor (x+1)/\eta \rfloor$. Let $Q(\mathbf{v} = (v_1, v_2, \ldots, v_d)) = (Q(v_1), Q(v_2), \ldots, Q(v_d))$
12: **return** $D_Q = \left\{ Q\left(\sqrt{\frac{d}{2}}\mathbf{v}_i\right) \right\}_{i=1}^q$,
13: **end procedure**

Algorithm 2: Computing the hash function $\mathfrak{h}_2(\cdot)$.

Description of the Protocol: We now describe our protocol in Algorithm 3 and the protocol parameters $\epsilon_v, \epsilon_{\ell,T}$ used. The protocol ensures two properties at the data owner:

Approximate Marginal Gain Computation: The trusted aggregator at the beginning (Step 4) shares $\tilde{\mathfrak{g}} = \mathfrak{h}_2(\mathfrak{h}_1(D_v))$. We show that $\|\mathfrak{h}_2(\mathfrak{h}_1(D_v)) - \sum_{\mathbf{x}\in D_v} \mathfrak{h}_1(\mathbf{x})\|_\infty$ is very small. Therefore, $\tilde{\mathbf{g}}^T \mathfrak{h}_1(\mathbf{y})$ when

computed at a data owner with a new point $\mathbf{y}$ approximates $\sum_{\mathbf{x}\in D_v} \mathfrak{h}_1(\mathbf{x})^T\mathfrak{h}_1(\mathbf{y})$. Similarly, over any other iteration ℓ (in Step 4), the hashed vector $\mathfrak{g}_\ell$ is such that $\mathfrak{g}_\ell^T\mathfrak{h}_1(\mathbf{y}) \approx \sum_{\mathbf{x}\in D_s} \mathfrak{h}_1(\mathbf{x})^T\mathfrak{h}_1(\mathbf{y})$. Since, $\mathfrak{h}_1$ has the property that $\mathfrak{h}_1(\mathbf{x})^T\mathfrak{h}_1(\mathbf{y}) \approx k(\|\mathbf{x}-\mathbf{y}\|)$, we can ensure that the maximization in Step 13 is approximately the marginal gain computation $J(D_s+\mathbf{y}) - J(D_s)$.

Differential Privacy: We also show that, due to application of $\mathfrak{h}_2$, all the releases seen by any data owner i are differentially private with respect to the current summary which also implies it is differentially private with respect to $\cup_{j\neq i} D_i - D_i$. Another key ingredient in our proof is in showing that the novel scheme in making the bid collection and winner notification process differentially private, while ensuring the parsimonious nature of the aggregator. Consider the Step 7 in Algorithm 3. Upon making a decision on the winning bid, the aggregator needs to acquire the winning point from the winner data source. Consider the following two naive ways of doing this: (a) Aggregator notifies the winner alone about the decision and acquires the data point. (b) Aggregator acquires data points from all the data sources. Keeping only the winner point, it discards the other points. An important observation here is that the first alternative is not differentially private. Indeed, it leaks information about the data points of the participating data sources. The second way is differentially private, indeed, each data source learns nothing new about other data sources. However, it is highly wasteful and contradicts the parsimonious nature of the aggregator. Indeed, in forming a summary of size p, it collects Kp data points. Our novel private auction (Steps 11-16 of Algorithm 3) obtains best of both scenarios, i.e., it is differentially private and accesses at most $O(pK^{\frac{1}{3}})$ data points in total.

1: **Input:** D_i $i \in [K]$, validation dataset D_v, seed set D_{init}, params $\{\epsilon_{auc}, \epsilon_v, \{\epsilon_{\ell,T}\}_{\ell=1}^p, \tau\}$.
2: **Output:** Summary D_s: $D_s \subseteq \cup_{i\in[K]} D_i$ such that $|D_s| = p$.
3: *Aggregator* initializes summary $D_s \leftarrow D_{\text{init}}$ and broadcasts $\tilde{\mathfrak{g}} = \mathfrak{h}_2(\mathfrak{h}_1(D_v), \epsilon_v)$.
4: **for** $\ell = 1 \dots p$ **do**
5: *Aggregator* broadcasts $\mathfrak{g}_\ell = \mathfrak{h}_2(\mathfrak{h}_1(D_s), \epsilon_{\ell,T})$.
6: Each *Data owner* $i \in [n]$ computes its "bid": $b_i = \max_{x\in D_i} \mathfrak{g}_\ell^T\mathfrak{h}_1(x) - \tilde{\mathfrak{g}}^T\mathfrak{h}_1(x)\frac{\ell}{\ell+1}$.
7: *Aggregator* chooses the best point through a private auction:
$x_{i^*} \leftarrow$ PRIVAUCTION$(b_i : i \in [n])$
8: *Aggregator* verifies the data point against the bid value and updates $D_s \leftarrow D_s \cup x_{i^*}$.
9: **end for**
10: **return** Summary $D_s - D_{\text{init}}$.

11: **procedure** PRIVAUCTION$(b_i : i \in [n])$
12: *Aggregator* orders the data owners, as $D'_1, D'_2, \dots, D'_K$, by their decreasingly bid values.
13: Independently with probability $\mathbb{P}[x_i] = e^{-\epsilon_{auc}(i-1)}$, *Aggregator* asks for the point x_i.
14: If a certain data point x was chosen τ times by a data source (Step 6), *Aggregator* asks for it.
15: *Aggregator* chooses a point x_*, with maximum bid value b_*, from the pool of all the points obtained so far and not yet included in the summary.
16: *Data owners* disconsider all the points sent to the *Aggregator* in the future iterations.
17: **end procedure**

Algorithm 3: Description of the protocol.

Theorem 2. *Let $a \in (0,1)$ and $\tilde{\delta} \in (0, 1/e)$ be any fixed constants. In Algorithm 3, for $|D_v| \geq \frac{44\sqrt{2}\sqrt{d}\log d \log^2 p}{\varepsilon_v}$, $|D_{\text{init}}| \geq 121 * 8d^2 \log^2 d \log(\frac{1}{\tilde{\delta}}) \log^5 p$, $d \geq \frac{16(\log 2N)(\log p)^2}{a^2}$, and setting $\eta \leq \frac{1}{d}$, $T = d^2$, $\varepsilon_v = \frac{\epsilon}{16T}$, $\varepsilon_{\ell,T} = \frac{\epsilon}{\sqrt{16T\ell\log(\frac{1}{\tilde{\delta}})\log p}}$, we obtain the following guarantees:*

(Differential Privacy) *Releases of the aggregator to any data owner i is $(\epsilon, \tilde{\delta})$-differentially private over all the iterations/epochs with respect to the datasets $\cup_{j\neq i} D_i$. Similarly, we have $(\epsilon, \tilde{\delta})$-differentially privacy over all the iterations w.r.t. validation set D_v.*

(Approximation Guarantee) *Let* OPT *denote an optimal summary set and D_s be the set of points obtained by Algorithm 3. We have $J(D_s) \geq (1-\frac{1}{e})J(\text{OPT}) - \Delta$, where $\Delta < O(\frac{\log p\sqrt{\ln d}}{\sqrt{d}}) + a + \frac{1}{\epsilon \log p} < 1$. Barring the Δ additive error the guarantees are close to the non-private greedy algorithm.*

(Parsimoniousness Guarantee) *Algorithm 3 is $O(\frac{\log\frac{1}{\delta}}{\epsilon}K^{\frac{1}{3}})$-parsimonious, i.e., in computing a summary of size p, it needs to access at most $O(pK^{\frac{1}{3}})$ data points.*

Differences between PRIVAUCTION and the Exponential Mechanism: There *may* be a superficial resemblance between Step 13 in the PRIVAUCTION procedure of Algorithm 3 and the exponential mechanism. Actually, our private auction is significantly different. First note that the probability of choosing the best bid is 1 which is not the case with the exponential mechanism. Secondly, while the exponential mechanism selects one approximately "best" point, we flip a coin for every bid whose bias has an exponentially decreasing relationship to the position of the bid in sorting order. Then, we choose multiple of them (instead of one) and a key proof point is to show that we can restrict the number of the points chosen overall. Finally, the bias probabilities do not even depend on the bid value (i.e., "score") while it would be the case for exponential mechanism.

Extension to a Less Trusted Curator. In our parsimonious curator model, the final summary dataset needs to be revealed to the trusted aggregator in order to train diverse models *downstream*. In Section F of the Appendix, we show that Algorithm 3 can be adapted to share just $\mathfrak{h}_1(\mathbf{x})$ hashes of data points. We show that this approach has some interesting privacy guarantees, specifically that the aggregator can only know the pairwise Euclidean distances between the points and nothing more. These hashes would be useful to train kernel based models such as *Support Vector Machines*.

4 Experimental Evaluation

We make an important observation that is crucial to obtain good performance in practice. According to Theorem 3, in order to control the additive error in approximating the query $\frac{w(D,i)}{q}$, Algorithm 2 needs: (a) T (the number of iterations) in Algorithm 2 to be larger than d^2 to match the distribution P_{avg} to the empirical distribution of coordinate i in the current summary D_s, (b) D_{init}, the size of the initial seed summary also needs to be large enough because of this (refer Theorem 2). Over multiple epochs of Algorithm 3 (Step 4) , we make the following changes to deal with these issues. *First Epoch ($\ell = 1$):* In practice, we 'seed' the protocol with a small initial seed set D_{init} to satisfy (b) and set $T = T_{\text{init}}$ to be large enough ($d^{1.5}$) to satisfy (a). *Subsequent Epochs ($\ell > 1$):* Clearly, the summary D_s grows and hence (b) is satisfied. We set $T = T_{\text{sub}}$ to be a constant for subsequent iterations. This may seem to contradict the requirement (a). However, we observe that $\mathfrak{h}_2(\cdot)$ operates on a summary that is only differing in one point from the previous iteration. Intuitively, a single point addition results in a small shift in the empirical distribution. Small incremental changes to the empirical distribution need to be matched incrementally. Thus, it is sufficient to have a significantly smaller number of iterations than that in Theorem 4. Therefore, T_{sub} is set to be small. We set the parameters of our algorithm as follows: the RBF kernel parameter $\gamma = 0.1$, dimension of Rahimi-Recht hash function $\mathfrak{h}_1(.)$ as $d = 140$. We use two different T parameters for different epochs given by $T_{\text{init}}(= T, \ell = 1) = d^{1.5} = 1656$ and $T_{\text{subs}}(= T, \ell > 1) = 5$. $\varepsilon_v = 0.01$ is the ϵ parameter for $\mathfrak{h}_2(\cdot)$ for the validation set and $\varepsilon_{\ell,T}$ is set for $\mathfrak{h}_2(\cdot)$ on summaries D_s over epochs ℓ as 0.05 for $\ell = 1$, $\frac{0.01}{\sqrt{pT_{\text{subs}}}}$ for $\ell > 1$.

Differential Privacy: An important observation here is that we do not need to preserve the privacy of the seed set, since it can be completely random. We now bound the differential privacy of our parameters with respect to both the consumer data and the summary data points. *Consumer Dataset (D_v):* We compute $\mathfrak{h}_2(\mathfrak{h}_1(D_v))$ only once i.e., in the first epoch. This involves $T_{\text{init}} = 1656$ iterations in Algorithm 2, with $\varepsilon_v = 0.01$. Applying Theorem 7 (in Appendix), we see that the total differential privacy measure $\varepsilon = 1.4$ (setting $\tilde{\delta} = 0.01$). *Summary Dataset (D_s):* Over p epochs of Algorithm 3, we have 5 iterations each with differential privacy $\frac{0.01}{\sqrt{pT_{subs}}}$. Thus, again by applying Theorem 7 (in Appendix), we obtain a total differential privacy of 0.043 (with $\tilde{\delta} = 0.0001$).

Experiments on Real World Datasets: We now back our theoretical results with empirical experiments. We compare three algorithms: a) *Non-Private Greedy*, where the aggregator broadcasts the (exact) average of the hashed summary set (i.e., $\frac{W(D_s,i)}{q}$) and hashed validation set (i.e., $\frac{W(D_v,i)}{m}$). This is equivalent to the approach of Kim et al. (2016). b) *Private Greedy*, which is the Algorithm 3 with parameters set as above. c) *Uniform Sampling*, where we draw equal number $\frac{p}{K}$ of required samples from each data provider to construct a summary of size p. We empirically show that *private greedy* closely matches the performance of *non-private greedy* even under the strong differential privacy constraints. For comparison, we show that our algorithm outperforms *uniform sampling*. The motivation for choosing the latter as a candidate comes from the typical manner of using stochastic gradient descent approaches such as Federated Learning [McMahan et al. (2016)] that perform

uniform sampling. We experiment with two real world datasets. We discuss one of them, which is based on an Allstate insurance dataset from a Kaggle (2014) competition. We show similar results for the MNIST dataset, that contains image data for recognizing hand written digits, in the Appendix G.

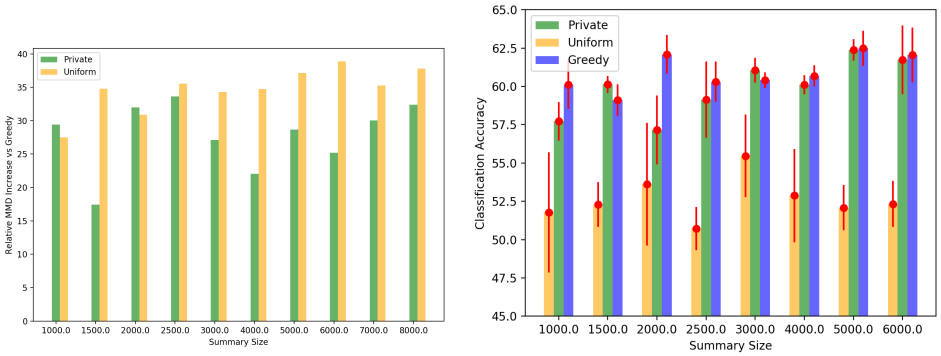

Figure 1: *All State Insurance Data:* (Top): Comparison of the percentage increase in MMD^2 of both the private and uniform sampling algorithms with respect to baseline greedy algorithm. Lower values indicate better performance. The private algorithm performs consistently better than uniform sampling. (Bottom): Comparison of the classification accuracy of the three algorithms using a Linear SVM classifier. Higher numbers indicate better performance. Our private algorithm outperforms uniform sampling by **6-10**% and closely matches the performance of the base line greedy algorithm.

All State Insurance Data: The dataset contains insurance data of customers belonging to different states of the U.S. The objective is to predict labels of one of the all-state products. In our setup, we use data corresponding to two states - *Florida* and *Connecticut*. We have four data owner participants, and an aggregator. The data is split up as follows: *Training data:* The training data is comprised of all the Florida data and 70% of the Connecticut data. The Florida data is split uniformly among the four data owners and Connecticut data is given to one of them. This allows us to create a skew in the data quality across different participants. *Validation data:* From the remaining 30% of Connecticut data we choose 25% of data as the validation data set. Note that we remove the labels from this validation set before giving it to the consumer. *Testing data:* The remaining Connecticut data is set aside as testing data. Thus the training data is solely comprised of Connecticut data. Further, we use around 150 points of random seed data belonging to a different state (*Ohio*). In our experiments, we vary the number of samples that need to be collected and compute the MMD^2 objective in each of these cases. In Figure 1, we compare the increase in MMD^2 with respect to greedy, i.e., $\frac{MMD^2(ALGM)-MMD^2(GREEDY)}{MMD^2(GREEDY)} \times 100$ where $ALGM$ is either our private greedy algorithm or the uniform sampling algorithm. Our results show that we consistently beat the uniform sampling algorithm while preserving differential privacy. In Figure 1, we compare the performance of these algorithms using a linear SVM. We find that the private algorithm while closely matching greedy beats uniform sampling by 6% to 10%.

5 Discussion

We consider a distributed data summarization problem in a transfer learning setting with privacy constraints. Different data owners have privacy constraints and a subset of points matching a target dataset needs to be formed. We provide a differentially private algorithm for this problem in the parsimonious curator setting, where the data owners do not wish to reveal information to other data owners and a curator entity can only access limited number of points.

Acknowledgement

We thank Naoki Abe and Michele Franceshini for helpful discussions in the initial stages of this work. We also thank anonymous reviewers for their thoughtful suggestions that helped improve our presentation of the paper.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318. ACM, 2016.

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pp. 464–473. IEEE, 2014.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 486–503. Springer, 2006.

Dwork, C., Nikolov, A., and Talwar, K. Using convex relaxations for efficiently and privately releasing marginals. In *Proceedings of the thirtieth annual symposium on Computational geometry*, pp. 261. ACM, 2014a.

Dwork, C., Roth, A., et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014b.

Ganin, Y. and Lempitsky, V. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

Giraud, B. G. and Peschanski, R. From" dirac combs" to fourier-positivity. *arXiv preprint arXiv:1509.02373*, 2015.

Gottlieb, J. and Khaled, R. Fueling growth through data monetization. https://www.mckinsey.com/business-functions/mckinsey-analytics/our-insights/fueling-growth-through-data-monetization, December 2017.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. J. A kernel method for the two-sample problem. *CoRR*, abs/0805.2368, 2008.

Hamm, J., Cao, Y., and Belkin, M. Learning privately from multiparty data. In *International Conference on Machine Learning*, pp. 555–563, 2016.

Hardt, M. and Rothblum, G. N. A multiplicative weights mechanism for privacy-preserving data analysis. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pp. 61–70. IEEE, 2010.

Hardt, M., Ligett, K., and McSherry, F. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems*, pp. 2339–2347, 2012.

Jukna, S. *Extremal combinatorics: with applications in computer science*. Springer Science & Business Media, 2011.

Kaggle. Allstate purchase prediction challenge. https://www.kaggle.com/c/allstate-purchase-prediction-challenge, 2014.

Kairouz, P., Oh, S., and Viswanath, P. The composition theorem for differential privacy. *IEEE Transactions on Information Theory*, 63(6):4037–4049, 2017.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

Kifer, D., Smith, A., and Thakurta, A. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pp. 25–1, 2012.

Kim, B., Khanna, R., and Koyejo, O. O. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*, pp. 2280–2288, 2016.

Mardia, K. V. and Jupp, P. E. *Directional statistics*, volume 494. John Wiley & Sons, 2009.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

Mitrovic, M., Bun, M., Krause, A., and Karbasi, A. Differentially private submodular maximization: Data summarization in disguise. In *International Conference on Machine Learning*, pp. 2478–2487, 2017.

Nemhauser, G. L., Wolsey, L. A., and Fisher, M. L. An analysis of approximations for maximizing submodular set functions—i. *Mathematical Programming*, 14(1):265–294, 1978.

Nissim, K. and Stemmer, U. Clustering algorithms for the centralized and local models. *arXiv preprint arXiv:1707.04766*, 2017.

Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., and Talwar, K. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

Pathak, M., Rane, S., and Raj, B. Multiparty differential privacy via aggregation of locally trained classifiers. In *Advances in Neural Information Processing Systems*, pp. 1876–1884, 2010.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Rubinstein, B. I., Bartlett, P. L., Huang, L., and Taft, N. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *Journal of Privacy and Confidentiality*, 4(1):65–100, 2012.

Sarpatwar, K. K., Ganapavarapu, V. S., Shanmugam, K., Rahman, A., and Vaculín, R. Blockchain enabled AI marketplace: The price you pay for trust. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 0, 2019.

Shokri, R. and Shmatikov, V. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pp. 1310–1321. ACM, 2015.

Song, S., Chaudhuri, K., and Sarwate, A. D. Stochastic gradient descent with differentially private updates. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pp. 245–248. IEEE, 2013.

Talwar, K., Thakurta, A. G., and Zhang, L. Nearly optimal private lasso. In *Advances in Neural Information Processing Systems*, pp. 3025–3033, 2015.

Thakurta, A. G. *Differentially private convex optimization for empirical risk minimization and high-dimensional regression*. The Pennsylvania State University, 2013.

Tzeng, E., Hoffman, J., Zhang, N., Saenko, K., and Darrell, T. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

Wu, X., Kumar, A., Chaudhuri, K., Jha, S., and Naughton, J. F. Differentially private stochastic gradient descent for in-rdbms analytics. *arXiv preprint arXiv:1606.04722*, 2016.