Aggregation for Sensitive Data

Avradeep Bhowmik
University of Texas at Austin
Austin, TX, USA

Joydeep Ghosh
University of Texas at Austin
Austin, TX, USA

Oluwasanmi Koyejo
University of Illinois at Urbana-Champaign
Urbana-Champaign, IL USA

Abstract—In many modern applications, considerations like privacy, security and legal doctrines like the GDPR put limitations on data storage and sharing with third parties. Specifically, access to individual level data points is restricted and machine learning models need to be trained with aggregated versions of the datasets. Learning with aggregated data is a new and relatively unexplored form of semi-supervision. We tackle this problem by designing aggregation paradigms that conform to certain kinds of privacy or non-identifiability requirements. We further develop novel learning algorithms that can nevertheless be used to learn from only these aggregates. We motivate our framework for the case of Gaussian regression, and subsequently extend our techniques to subsume arbitrary binary classifiers and generalised linear models. We provide theoretical results and empirical evaluation of our methods on real data from healthcare and telecom.

I. Introduction

While most machine learning paradigms assume constant training-time access to a full set of raw non-aggregated "individual level" data points (e.g., individual rows of a data table), this is not a feasible scenario in many situations involving sensitive data – instead, the data is only accessible to a learner for a limited period of time, after which it must be stored in an aggregated form (e.g., column averages for the same data table). This is a common scenario in many technology driven domains like the telecommunication industry, where legal requirements (e.g. FTC rules [4]) mandate the erasure of individual consumer data after a specified period. Along similar lines, privacy considerations in domains like healthcare [13] limit dissemination of patient records to third parties. Finally, when data needs to be transferred over an expensive or less secure channel, it is preferable to summarise the data before transfer[15].

In all such cases, any machine learning solution has to operate under the constraint that data will only be available in small subsets for brief periods, after which they need to be aggregated by the system, and the original dataset purged in its entirety. Existing work from privacy[6, 7], sketching [12] or online learning [3] cannot be applied directly to our setup, since they require repeated data access or modify the relationship between observed variables. At the other end of the pipeline, ecological fallacy concerns put up additional challenges in learning from aggregated data [2, 14].

Our work is motivated by two complementary objectives:

1) design aggregation paradigms that protect data security and privacy
2) formulate learning algorithms that can use these aggregates or summaries to train predictive models that are effective at individual level predictions

This is a tall task, but we show that both these objectives can be achieved in several cases. To illustrate our ideas, consider the case of Gaussian regression where covariates $\mathbf{X}$ are related to targets $\mathbf{y}$ via a linear parameter $\boldsymbol{\theta}$ as $\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2)$ It is well known that the MLE parameter $\boldsymbol{\theta}_{MLE}$ can be obtained in closed form from the data as

$$\boldsymbol{\theta}_{MLE} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$$

Clearly, we do not require the entire dataset – the only relevant quantities that are required for learning the model are aggregates $\mathbf{X}^\top\mathbf{X}$ and $\mathbf{X}^\top\mathbf{y}$.

Therefore, if we store only these aggregates, and delete the individual datapoints themselves, we can still recover the MLE parameter error-free without access to the raw dataset. For the Gaussian case, therefore, we have an exact solution. We can use this as a basic modus operandi for other setups as well. Our specific contributions are summarised below:

1) We design novel aggregation paradigms and learning algorithms that guarantee privacy while still allowing learning for a wide variety of models.
2) We provide a theoretical analysis as well as empirical evaluation for our methods with experiments on data from telecommunication and healthcare

We call our framework SlAgg, or Slice and Aggregate, after the main steps involved in the procedure. While keeping the overall approach fairly simple and intuitive, we prove strong guarantees on its performance, and also show very favorable empirical results.

II. Problem Definition

In this work, we consider predictive models that are trained via supervised learning. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_N] \in \mathbb{R}^{N \times d}$ be a set of N data points in a d-dimensional feature space, and let $\mathbf{y} = [y_1, y_2, \cdots y_N] \in \mathcal{Y}^N \subseteq \mathbb{R}^N$ be their corresponding targets. We assume that their exists a function f such that for each $(\mathbf{x}, y)$ pair, we have $y = \mathrm{f}(\mathbf{x}) + \eta$, where η is random noise.

The standard machine learning setup estimates this function f using a training set of the form $\mathcal{D} = (\mathbf{X}, \mathbf{y}) \equiv$

$\{(\mathbf{x}_i, y_i) : i = 1, 2, \cdots\}$ and a learning protocol that consists of solving the following optimisation problem

$$\mathrm{f}^* = \arg\min_{\mathrm{f} \in \mathcal{F}} \sum_{(\mathbf{x},y) \in \mathcal{D}} \mathcal{L}(\mathrm{f}(x), y) \tag{1}$$

where $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}^+$ is a loss function that measures the discrepancy between predicted $\mathrm{f}(\mathbf{x})$ and measured y (e.g., negative log-likelihood).

In our setup, the full dataset is not available for training. Instead, the data is divided into M disjoint "chunks" or subsets as $\mathcal{D}_T = \{(\mathbf{x}_i, y_i) : i \in \mathcal{I}_T\}$, where $\mathcal{I}_T \subset [N]$ are partitions of the index set, and $T = 1, 2, \cdots M$. For example, $\mathcal{D}_T$ may be customer data or patient records for the T^{th} month, which need to be compiled into non-identifiable aggregates and the individual data points are to be deleted at the beginning of the $(T+1)^{th}$ month for privacy reasons.

Therefore, instead of the full dataset $\mathcal{D}$, the learner is only allowed access to each chunk, one at a time, for a brief period of time. The learner's task is to use these chunks to learn a non-identifiable aggregates before the individual data points in each chunk are deleted. Finally, the learner will be required to devise a training algorithm for the final predictive model that only use these aggregates.

III. Aggregation Design Paradigms

The question now is how to use these chunks to learn an effective estimate of the function f. For this, we take inspiration from the concept of "sufficient statistics" in estimation theory that studies various methods to estimate a parameter for a distribution given data. Let $\boldsymbol{\theta}$ be a parameter to be estimated from a given dataset $\mathcal{D}$. A sufficient statistic for $\boldsymbol{\theta}$ is a quantity (or a set of quantities) $\mathcal{S}$ computed from the dataset $\mathcal{D}$ such that the posterior of the parameter given the statistic is independent of the individual datapoints themselves, that is, $P(\boldsymbol{\theta}|\mathcal{S}, \mathcal{D}) = P(\boldsymbol{\theta}|\mathcal{S})$. Basically, a sufficient statistic summarises the dataset by extracting from the individual datapoints all the information that is necessary for parameter estimation, and discards the rest.

Our task here is similar – given a data chunk, extract the useful information from the data chunk in the form of aggregates that can be subsequently used for training a final predictive model. We now discuss specific instantiations of both an aggregation paradigm as well as a learning algorithm that only uses these aggregates. We have already seen this idea in action for the case of Gaussian linear regression. In the rest of the manuscript, we extend these methods to the case of binary classification and generalised linear models.

A. Binary Classifiers

Unlike the Gaussian case, there is no nice closed form solution for most binary classification models. In fact, the model parameter itself may not always be unique and suffer from identifiability issues owing to rotational or scale invariance. Therefore, we study the case of binary classification not in formal model specification terms, but by treating a classifier as a black box with a specific probability of error over the population.

In particular, consider the case where one has access to multiple noisy classifiers. One can consider the output of each of these classifiers as noisy estimates for the "true" class label (defined as the mode of $P(y|x)$), and by taking the majority vote, one can estimate the true class label with high accuracy. Therefore, if we can "aggregate" each data chunk to learn a black box noisy binary classifier, we no longer need individual training datapoints themselves to get the final predictive model.

Hence, our protocol is the following:

1) For each data chunk $\mathcal{D}_T$, learn a classifier $\mathrm{f}_T : \mathcal{X} \mapsto \{0, 1\}$ from only the data points in $\mathcal{D}_T$
2) Given a new random sample $\mathbf{x}$, and the classifiers $\{\mathrm{f}_T : T = 1, 2, \cdots M\}$, obtain the corresponding predictions $\{\widehat{y}_T = \mathrm{f}_T(\mathbf{x}) : T = 1, 2, \cdot M\}$
3) Obtain the final estimate for the class label as

$$\widehat{\mathrm{f}(\mathbf{x})} = \mathrm{median}\{y_T : T = 1, 2, \cdots M\} \tag{2}$$

We now analyse the predictive accuracy of our final classifier. To account for unavoidable noise and limitations of model class, we compare the performance of our method to the best possible model from the function class that can be learned from the individual non-aggregated data points. Let λ be the probability of mis-classification on a randomly selected data point for the best possible model f^* from the function class. For any $\mathbf{x}$, let $z_T(\mathbf{x}, y) = \mathbb{I}\{\mathrm{f}_T(\mathbf{x}) \neq y\}$ where $\mathbb{I}$ is the indicator function. Note that since each data chunk $\mathcal{D}_T$ consists of i.i.d samples of the same size, z_T are independent and identically distributed random variables over the joint probability space for the data. Let $p = E[z_T]$ be an upper bound on the mis-classification probability for the T^{th} classifier, with the expectation taken over the joint distribution of $(\mathbf{x}, y)$. We then have the following result:

Proposition III.1. Let $p < 0.5$ and $\widehat{\mathrm{f}}$ be our final classifier from M data chunks as defined in equation (2). Then, the probability that $\widehat{\mathrm{f}}$ does worse than f^* on any given datapoint is upper bounded by the quantity:

$$\frac{1-p}{(1-\lambda)(1-2p)} \left[(1-p)p \ exp\left(2\kappa - \xi_M + \zeta_M\right))\right]^{M/2}$$

where $\kappa \approx 0.693$, $\xi_M \sim O(\frac{logM}{M})$, and $\zeta_M \sim O(\frac{1}{M^2})$

It is easy to see that as M increases, the probability of error rapidly decreases. Note that one corollary of this result is that a learner that uses data chunks can potentially learn better than a single learner that uses the full non-aggregated dataset. Indeed, this is exactly what happens with our experiments on real data as we show in section (IV).

Multi-Class Case: Our analysis extends to the multi-class case by treating it as multiple 2-class classification, and then using union bound to get an upper bound on error. A similar result holds as above, with an additional multiplicative cost factor, which can be tuned up to certain trade-offs.

B. GLMs and Exponential Family Distributions

We now extend our techniques to GLMs [1] which are generalizations of linear regression that subsume various models like Poisson regression, logistic regression, etc. as special cases. A GLM is usually parametrized by a convex function $\boldsymbol{\phi}$ (usually known[1]) and a parameter $\boldsymbol{\theta}$ (to be learned from data). Given a predictor $\mathbf{x}$ and a parameter $\boldsymbol{\theta}$, a GLM generates the target y from a linear function of the predictor $\mathbf{x}^\top\boldsymbol{\theta}$ using a monotonic link function $\mathbf{g}_{\boldsymbol{\phi}}(\cdot)$ using a probability distribution $P_{\boldsymbol{\phi}}$ from the exponential family. Specifically, we have

$$P_{\boldsymbol{\phi}}(y|\mathbf{x},\boldsymbol{\theta}) \propto \ exp\left(y\mathbf{x}^\top\boldsymbol{\theta} - \mathbf{G}_{\boldsymbol{\phi}}(\mathbf{x}^\top\boldsymbol{\theta})\right) \tag{3}$$

where $\mathbf{G}_{\boldsymbol{\phi}}$ is such that $\mathbf{g}_{\boldsymbol{\phi}} \equiv \nabla\mathbf{G}_{\boldsymbol{\phi}}$. The specific $P_{\boldsymbol{\phi}}$ depends on the GLM used (e.g. Poisson for Poisson regression, Bernoulli for logistic regression, etc.).

Unbiased Estimators Generally speaking, learning the MLE parameter $\boldsymbol{\theta}^*$ for a GLM from anything other than individual data points can be difficult. However, if we have access to unbiased estimates of $\boldsymbol{\theta}^*$, we can still approximate the model parameter by averaging. Let $\mathbb{P}$ be an unbiased estimator that takes any dataset $\mathcal{D}$ and outputs an estimate for the parameter $\mathbb{P}(\mathcal{D})$ such that $\forall\mathcal{D},\ E_{\mathcal{D}}[\mathbb{P}(\mathcal{D})] = \boldsymbol{\theta}^*$, the optimal model parameter. We partition the data into chunks $\mathcal{D}_T$, define $\widehat{\boldsymbol{\theta}}_T = \mathbb{P}(\mathcal{D}_T)$ as result of the estimator applied to the data chunk. We define our final parameter estimate as $\widehat{\boldsymbol{\theta}} = \frac{1}{M}\sum_{T=1}^{M}\widehat{\boldsymbol{\theta}}_T$ It is easy to see that if M is high enough, then with high probability, $\widehat{\boldsymbol{\theta}} \to \boldsymbol{\theta}^*$, even if the individual $\widehat{\boldsymbol{\theta}}_T$ be of low quality.

Setting the gradient of the log likelihood with respect to $\boldsymbol{\theta}$ to zero gives $\mathbf{X}\mathbf{g}_{\boldsymbol{\phi}}(\mathbf{X}^\top\boldsymbol{\theta}) = \mathbf{X}\mathbf{y}$ where $\mathbf{g}_{\boldsymbol{\phi}}(\cdot)$ is applied elementwise. Clearly, this does not have a closed form solution except when $\mathbf{X}, \mathbf{X}^\top$ are both invertible. Suppose we divided up $\mathcal{D}$ in chunks of d data samples each, where d is the dimensionality of the data. Then for each T, we can obtain a parameter $\widehat{\boldsymbol{\theta}}_T$ that is locally optimal for the samples corresponding to the data chunk $\mathcal{D}_T$.

Therefore, our learning protocol can be summarised as follows –

1) For each data chunk $\mathcal{D}_T$, compute a locally optimal parameter as
$$\widehat{\boldsymbol{\theta}}_T = (\mathbf{X}_T\mathbf{X}_T^\top)^{-1}\mathbf{X}_T\mathbf{g}_{\boldsymbol{\phi}}^{-1}(\mathbf{y}_T)$$
2) Using the individual $\widehat{\boldsymbol{\theta}}_T$ for each data chunk $\mathcal{D}_T$, compute the final estimate for the global GLM parameter as
$$\widehat{\boldsymbol{\theta}} = \frac{1}{M}\sum_{T=1}^{M}\widehat{\boldsymbol{\theta}}_T$$

Here, $\mathbf{g}_{\boldsymbol{\phi}}^{-1}$ is defined element-wise. In case y is outside the domain of $\mathbf{g}_{\boldsymbol{\phi}}^{-1}$, one can use any projection of y to the interior o the domain of $\mathbf{g}_{\boldsymbol{\phi}}^{-1}$ instead. We have the following result:

Proposition III.2. If $\mathbf{g}_{\boldsymbol{\phi}}^{-1}$ (equivalently $\mathbf{g}_{\boldsymbol{\phi}}$) is a linear function, $\widehat{\boldsymbol{\theta}}$ is an unbiased estimator of $\boldsymbol{\theta}^*$

The link function is effectively linear for many exponential family distributions, like Gaussian, Exponential, Pareto, Chi-Squared, etc., but one can use sampling-based approximations to estimate $\boldsymbol{\theta}^*$ for other GLMs.

IV. Experiments

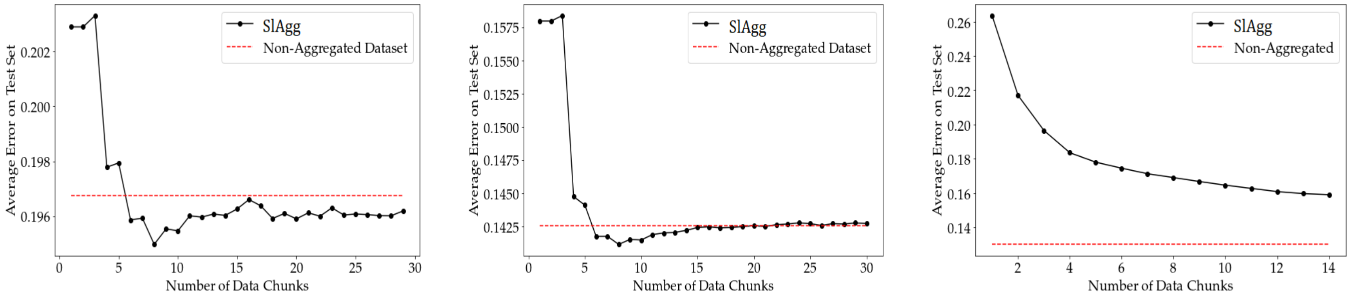
We demonstrate the efficacy of our methods with empirical evaluation on three real datasets from the healthcare and telecom domains where our problem setup is particularly relevant. Since this is a first work, we do not know of any alternative algortihmic competitors for our methods. Hence, in each case, we compare against a performance "upper-bound" that is obtained from learning from the full non-aggregated dataset with individual level samples. We also compare against SGD and an ecological regression (EcoReg) baseline [11] that treats averages as individual level samples for training. We show plots of test error versus number of learners/data chunks seen by our method, as well as a final table of results. SGD and EcoReg are included only in the table and omitted from plots for clarity, since their performance is rather poor in comparison.

Binary Classification: Churn in Telecom:

We use two datasets from the Telecom industry for our binary classification tasks one from IBM Watson Analytics, and the other from Kaggle. In both cases, the objective is to predict churn [8] from customer account and usage details, which refers to the event where a customer terminates a service or contract with a particular company. We use a logistic regression model as our base modelling framework, and use available information like demographic or service details as features (see [9, 10] for more details. For both datasets, our algorithm needs only a few data chunks to achieve a performance better than learner with non-aggregated dataset, and significantly outperforms SGD and EcoReg (Fig (1)).

Real-valued data: Healthcare

We now apply our methods on a healthcare dataset where the objective is to estimate Medicare charges from the CMS Beneficiary Summary DE-SynPUF dataset [5] with available predictor variables that include age, race, sex, duration of coverage, presence of a variety of chronic conditions, etc. This application is motivated by patient privacy considerations that limit access to healthcare records.The data is collected into chunks and fed into each algorithm to learn a Poisson regression model. The results (Fig (1) and Table (IV)) show that our techniques



(a) Test Error on IBM (churn) dataset (b) Test Error on Kaggle (churn) dataset (c) Test Error on DESynPUF

Fig. 1: Test error vs Number of Data Chunks on IBM, Kaggle and DESynPUF datasets

Note 1: Our algorithm does better than a binary classifier trained with non-aggregated data, exactly as predicted by Prop (III.1).

	No. of	Non-Aggregated		This Work		SGD		EcoReg	
	Chunks	Train	Test	Train	Test	Train	Test	Train	Test
IBM	29	0.1967	0.197	0.1947	0.196	0.285	0.287	0.356	0.357
Kaggle	30	0.142	0.142	0.142	0.143	0.245	0.245	0.216	0.218
DESynPUF	14	0.125	0.130	0.153	0.159	1.785	1.797	0.22	0.23

TABLE I: Final Training and Test Error on all three datasets for learner with non-aggregate data, our method with all chunks used, SGD and naive averaging. Our method outperforms baseline and has performance very close to learner with full, non-aggregated dataset. Note: We use logistic regression as base model for churn datasets, and Poisson regression for DE-SynPUF

with only a few data chunks can perform very close to a learner with access to the full dataset.

V. Conclusion

In this manuscript we tackle the problem of learning in the scenario when privacy, scalability, security, etc. concerns limit access to training data only in the form of chunks that need to be aggregated and deleted after a specific duration of time. We design aggregation techniques, as well as algorithms to learn models from these aggregates that can nevertheless make effective predictions at the individual level. We motivate our techniques by using Gaussian regression, and subsequently extend them to the case of binary classification and GLMs. We provide both theoretical results as well as empirical evaluation for our work.

References

[1] Arindam Banerjee, Srujana Merugu, Inderjit S Dhillon, and Joydeep Ghosh. 2005. Clustering with Bregman divergences. The Journal of Machine Learning Research 6 (2005), 1705–1749.

[2] Avradeep Bhowmik, Joydeep Ghosh, and Oluwasanmi Koyejo. 2015. Generalized Linear Models for Aggregated Data. In Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics. 93–101.

[3] Léon Bottou. 2010. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT'2010. Springer, 177–186.

[4] Federal Trade Commission. 2005. FACTA Disposal Rule Goes into Effect June 1.

[5] DESynPUF. 2008. Medicare Claims Synthetic Public Use Files (SynPUFs). Centers for Medicare and Medicaid Services (2008). http://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/index.html.

[6] Cynthia Dwork. 2008. Differential privacy: A survey of results. In International Conference on Theory and Applications of Models of Computation. Springer, 1–19.

[7] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science 9, 3–4 (2014), 211–407.

[8] Shin-Yuan Hung, David C Yen, and Hsiu-Yu Wang. 2006. Applying data mining to telecom churn management. Expert Systems with Applications 31, 3 (2006), 515–524.

[9] IBM. . Using Customer Behavior Data to Improve Customer Retention. IBM TJ Watson ( ). https://www.ibm.com/communities/analytics/watson-analytics-blog/predictive-insights-in-the-telco-customer-churn-data-set/.

[10] Kaggle. . Churn in Telecom dataset. ( ). https://www.kaggle.com/becksddf/churn-in-telecoms-dataset.

[11] Gary King, Martin A Tanner, and Ori Rosen. 2004. Ecological inference: New methodological strategies.

Cambridge University Press.

[12] Edo Liberty. 2013. Simple and deterministic matrix sketching. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 581–588.

[13] Yubin Park and Joydeep Ghosh. 2014. LUDIA an aggregate-constrained low-rank reconstruction algorithm to leverage publicly released health data. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 55–64.

[14] Novi Quadrianto, Alex J Smola, Tiberio S Caetano, and Quoc V Le. 2009. Estimating labels from label proportions. The Journal of Machine Learning Research 10 (2009), 2349–2374.

[15] David Wagner. 2004. Resilient aggregation in sensor networks. In Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks. ACM, 78–87.