# EstImAgg: A Learning Framework for Groupwise Aggregated Data

Avradeep Bhowmik * Minmin Chen † Zhengming Xing ‡ Suju Rajan §

**Abstract**

Aggregation is a common technique in data-driven applications for handling issues like privacy, scalability and reliability in a vast range of domains including healthcare, sensor networks and web applications. However, despite the ubiquitousness, extending machine learning methods to the aggregation context is unfortunately not well-studied. In this work, we consider the problem of learning individual level predictive models when the target variables used for training are only available as aggregates. In particular, this problem is a critical bottleneck in designing effective bidding strategies in the context of online advertising where ground-truth cost-per-click (CPC) data is aggregated before being released to advertisers. We introduce a novel learning framework that can use aggregates computed at varying levels of granularity for building individual-level predictive models. We generalise our modelling and algorithmic framework to handle data from diverse domains, and extend our techniques to cover arbitrary aggregation paradigms like sliding windows and overlapping/non-uniform aggregation. We show empirical evidence for the efficacy of our techniques with experiments on both synthetic data and real data from the online advertising domain as well as healthcare to demonstrate the wider applicability of our framework.

## 1 Introduction and Related Work

Data aggregation is a ubiquituous practice in a wide variety of domains[25, 20, 23] for handling issues like privacy, scalability and reliability. In industry, for example, aggregation is a common technique to preserve the integrity of proprietary data when two or more different companies have to work on a common platform to provide a service or a product to consumers. In such situations, it is very common for individual companies to protect proprietary ownership by only sharing their data with third parties in an aggregated form.

Online advertising [12, 27] is a domain where this scenario arises all the time. The advertising process [34, 32] consists of two main players– the advertiser (e.g. Criteo or AdRoll) who provides the ads and bids for advertising space, and the publisher (e.g. Google or Facebook) who provides the platform on which to display such ads. The highest bidder is allowed to place an ad of their choice on the provided platform, and charged a specific fee whenever an end-user clicks on the ad. These transactional charges (called cost-per-click or CPC) are based on a pre-negotiated but only partially declared auction mechanism that depends, among other things, on the advertiser's bid, the position of the ad, demand etc. Modelling and estimating CPC beforehand, therefore, is a key component of the complex bidding strategy [11, 33], and also in the decision-making process for selecting the specific advertisement that can optimally balance critical trade-offs between cost incurred and potential revenue earned from a given ad-space.

Unfortunately, for various reasons like transactional efficiency, protection of proprietary mechanism design, etc. publishers often submit the invoice of charges to advertisers on a cumulative basis (total charges over a day, etc.) rather than on a per-click basis. Therefore, the transaction data available to the advertiser for training their models only consists of aggregated CPC values, where the per-click charges have been obfuscated through averaging. A specific example of this type of data sharing happens on the Google Shopping product [21] where the platform shares only the aggregated and not per click cost information. Developing a framework to address this use case– training individual-level models with aggregated data– is thus a critical bottleneck in online advertising.

While we use online advertising as a motivating example, the specific setup we consider is ubiquitous across a much wider variety of domains. Aggregation is used as a statistical disclosure limitation technique in many privacy sensitive domains like healthcare [25] where confidential information like hospital records are often aggregated to protect individual patients' privacy. In large scale data collection settings like census or population surveys [20] or meteorological studies [16], individual level data is often collected or stored as

*The University of Texas at Austin, Austin, TX
†Google, Mountain View, CA
‡Criteo, Palo Alto, CA
§Criteo, Palo Alto, CA

aggregates for scalability reasons. Sensor networks and IoTs [31] use data aggregation in the interest of robustness, when measurements by individual sensors tend to be corrupted with noise that gets canceled out when the measurements are averaged over space or time. Financial forecasting applications depend on economic metrics which are often released as aggregates [23, 22] by governmental agencies and independent think tanks.

The key problem we focus on in this work is this– how do we build models that work at the individual level but that can nevertheless be trained with data collected at the aggregate level? Unfortunately, despite its near universal presence, learning from aggregated data is still a relatively unexplored topic, and there are rarely any easy answers. This is a new and extremely challenging semi-supervised learning paradigm, and naive application of standard techniques almost always fail because of the ecological fallacy[29], wherein inferences drawn at the group level are significantly different from those drawn at the individual level.

In this manuscript, we introduce a novel modelling and algorithmic framework to learn individual level models for the case when the target variables of interest are collected into group-wise aggregates. We emphasise that we use online advertising only as an example application for easier presentation of our learning framework. Our methods are extremely general and can be used for any application that involves learning from aggregated target variables. To this end, we use as our base modelling framework generalised linear models (GLMs), which are a large class of models that can handle diverse data types (real valued, binary, integer, etc.) and are the primary work-horse in a vast range of domains, from climate science to recommendation engines to healthcare. Since nearly all existing work on generalised linear models assume access to individual level data, we introduce significant modelling innovations on top of existing theory and algorithms in prior art that enables us to extend this large class of models to the aggregated data scenario.

**Contributions:** Our specific contributions are outlined below–

1. We introduce a novel framework that can learn individual level generalised linear models when the target data is available only as aggregates computed over sub-groups of the data space. To the best of our knowledge, we are the first to tackle this problem.

2. We design a new learning algorithm that uses alternating data imputation and estimation steps to train the generalised linear model with access only to aggregated target variables

3. We extend our analysis to cover cases where the data aggregation has been performed over arbitrary grouping paradigms to subsume cases like overlapping aggregation, sliding window, non-uniform aggregation, etc.

4. We empirically evaluate our methods on both synthetic and real datasets from the advertising domain to show the efficacy of our techniques. We further demonstrate the general applicability of our methods by evaluating the performance of our techniques on problems from the healthcare domain.

Learning from aggregated data is a fairly new and unexplored area and prior literature is extremely limited in both scope as well as applicability. A similar problem in the classification literature is learning from label proportions (LLP) [28, 26] which involves estimation of classifiers given only the proportion of discrete valued labels in groups or bags of labeled targets. Unlike their work which is restricted to the classification setting where the target variables can only take values from a finite alphabet, our work concerns itself with general regression where the target variable can take arbitrary real values. The problem of reconstructing individual level records using sample mean data was been studied in [24] and [25], but unlike our work, they focused on data reconstruction rather than predictive modelling, and assumed a low rank structure and a Gaussian model for their data matrix. Authors in [4] introduced and empirically evaluated an estimation algorithm for regression involving histogram aggregation– our work looks at group-wise means instead, which lacks the richness of structure that the histogram setting provides. Linear regression involving group-wise aggregates was studied in [5], but they required restrictive assumptions like sparsity for the parameter vector, and restricted isometry properties for the feature matrix. Another work [6] used Fourier analysis for simple linear regression with aggregated spatio-temporal data, but made assumptions like signal stationarity. Unlike the aforementioned papers, our work needs no data generation assumptions (isometry, low rank, sparsity) and can handle linear regression (with a Gaussian model) as well as the much larger class of GLMs.

Finally, our base modelling framework for our applications of interest are generalized linear models (GLM) [19] which are a generalization of linear regression that subsumes various models like Poisson regression, logistic regression, etc. as special cases. The specific properties of GLMs that we shall use in our work shall be described

in section 3.1. We also refer the reader to [18] or [17] for a detailed discussion on GLMs.

## 2 Problem Description

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_n]$ be a set of feature vectors for $n$ data samples and $Y = [y_1, y_2, \cdots y_n]$ be their corresponding values for the target variables of interest. In the interest of clarity and notational succinctness, we start off by describing our framework within the simplified context of linear models, and extend our analysis to the more general version in section 3. For a linear model, the corresponding target $y$ for each covariate $\mathbf{x}$ is generated via a vector parameter $\boldsymbol{\beta}$ as

$$y = \mathbf{x}^\top \boldsymbol{\beta} + \epsilon$$

where $\epsilon$ is a zero mean noise term. Maximum likelihood parameter estimation involves solving an optimisation problem to minimise the regularised negative log-likelihood of the observed data,

$$\boldsymbol{\beta}^* = \arg\min_{\boldsymbol{\beta}} \sum_{(\mathbf{x},y)\in(\mathbf{X},Y)} \mathcal{L}(\mathbf{x}^\top\boldsymbol{\beta}, y) + \lambda\mathcal{R}(\boldsymbol{\beta}) \tag{2.1}$$

where $\mathcal{L}$ is the loss function ($\mathcal{L} = \|\cdot\|^2$ for Gaussian noise) and $\mathcal{R}(\cdot)$ is an appropriate regulariser ($\ell_2$ for ridge regression, $\ell_1$ for LASSO, etc.).

In the standard regression setting, the data used for training the model is available at the individual level as (feature, target) pairs $\mathbb{D}_{(x,y)} = \{(\mathbf{x}_i, y_i) : i = 1, 2, \ldots n\}$. In our case, we do not have access to data at this level of granularity– instead, while $\mathbf{X}$ is fully observed, the target values $\{y_i : i = 1, 2, \cdots n\}$ are subjected to an aggregation process (partially specified, and not controlled by the learning agent) that produces a set of $m$ summaries $\{z_k : k = 1, 2, \cdots m\}$ that are then made available to us. Specifically, we are provided with summaries $z_k = \sum_{i\in\mathcal{I}_k} y_i$, where each $\mathcal{I}_k \subset \{1, 2, \cdots m\}$ is an index set that defines which target variables contribute to a given aggregate.

In the online advertising example, each $y_i$ may be the individual CPC and $z_k$ is the aggregate CPC value computed over all click activity in the $k^{th}$ aggregation window, while $\mathbf{x}_i$ may be information like that is available in full to the advertiser like price or brand for products, or target country, campaign type, etc. for the ad itself. In privacy-sensitive applications, $z_k$ might refer to aggregated information like health metrics or income aggregated at the zipcode level, while $\mathbf{x}_i$ may refer to data like demographic information (race, gender, etc.) available publicly at the individual level from, say, voter or census files. We assume for now that the aggregation indices are disjoint, that is, $\mathcal{I}_k \cap \mathcal{I}_{k'} = \varphi$ for $k \neq k'$, we extend this to the overlapping case in section 3.2.

We now introduce our learning framework for $\boldsymbol{\beta}$ that can predict individual level targets $\hat{y}$, but use only the aggregates $z$ for training. Let $\mathbf{X} \in \mathbf{R}^{n\times d}$ be the matrix of $n$ feature vectors and $\mathbf{z} \in \mathbf{R}^m$ be the vector of aggregates. For notational succinctness, we use $\mathbf{y} \in \mathbf{R}^n$ to denote imputed or predicted targets, and use $\mathbf{y}_{true} \in \mathbf{R}^n$ for the "ground truth". We call our framework EstImAgg or Estimate-IMpute for AGGregated Data after the main steps involved in the learning process.

**Algorithm 1** EstImAgg: Basic
Non-overlapping aggregation, Gaussian model

Input: $\mathbf{X}, \mathbf{z}$, Aggregation groupings $\mathcal{I}_k$
Initialise $\{y_i = z_k : i \in \mathcal{I}_k\}$
**while** not converged **do**
  $\boldsymbol{\beta}^+ = \arg\min_{\boldsymbol{\beta}} \sum_{(\mathbf{x},y)} \|\mathbf{x}^\top\boldsymbol{\beta} - y\|^2 + \lambda\mathcal{R}(\boldsymbol{\beta})$
  **for** each $k \in \{1, 2, \cdots m\}$ **do**
    $\gamma_k \leftarrow \frac{1}{|\mathcal{I}_k|}\left(z_k - \sum_{i\in\mathcal{I}_k} \mathbf{x}_i^\top\boldsymbol{\beta}^+\right)$
    $y_i^+ \leftarrow \mathbf{x}_i^\top\boldsymbol{\beta}^+ - \gamma_k \quad \forall \quad i \in \mathcal{I}_k$
  **end for**
  Update variables $(\mathbf{y}, \boldsymbol{\beta}) \leftarrow (\mathbf{y}^+, \boldsymbol{\beta}^+)$
**end while**
**return** $\mathbf{y}, \boldsymbol{\beta}$

In the standard case, learning the model effectively implies minimising the loss function only over the parameter $\boldsymbol{\beta}$, as in Equation 2.1. In our case, we not only have to estimate the parameter $\boldsymbol{\beta}$, but also the non-aggregated targets $\mathbf{y}$ subject to the constraints that the imputed estimates agree with the aggregates $\mathbf{z}$, which adds an extra set of constraints $\sum_{i\in\mathcal{I}_k} y_i = z_k \quad \forall\ k \in \{1, 2, \cdots m\}$ to the optimisation problem 2.1.

We solve this using alternating minimisation. The first step, solving for $\boldsymbol{\beta}$ given a particular value of $\mathbf{y}$, is a simple regression parameter estimation problem. The second step, solving for $\mathbf{y}$ given a particular estimate for $\boldsymbol{\beta}$, is more interesting. For a Gaussian model ($\mathcal{L} = \|\cdot\|^2$), the optimisation problem is as follows

$$\mathbf{y}^+ = \underset{\mathbf{y}}{\text{argmin}} \sum_{(\mathbf{x},y)\in(\mathbf{X},\mathbf{y})} \|\mathbf{x}^\top\boldsymbol{\beta} - y\|^2 \quad \text{s.t.} \sum_{i\in\mathcal{I}_k} y_i = z_k \quad \forall\ k \tag{2.2}$$

By using standard optimisation theory (details skipped for lack of space) it can be proved that the optimal value for $\mathbf{y}$ for this problem can actually be obtained in closed form. As shown in Algorithm 1, the solution involves applying an additive update to each estimated target to make it compatible with the aggregates. The learning steps described in plain English above is summarised as Algorithm 1.

## 3 General Formulation

In this section, we generalise our framework to handle a wider class of problems by extending our methods

to generalised linear modelling, and modifying our algorithm to handle arbitrary aggregation paradigms.

**3.1 Generalised Linear Models** While least squares regression is useful for modeling continuous real valued data generated from a Gaussian distribution, this is not always a valid assumption. In many cases, the data of interest may be binary valued or count valued, and generalised linear models or GLM's are more appropriate for such scenarios– Gaussian models for real-valued, Poisson regression for integer valued, logistic for binary, log-Normal for non-negative reals, etc. GLM's have been successfully deployed in a wide variety of fields including healthcare [10], ecological surveys [13], medical trials [9], etc. and are one of the most common data science workhorses in almost any domain.

A generalized linear model generalises linear regression and subsumes various models like Poisson regression, logistic regression, etc. as special cases. A GLM, parametrized by a convex function $\boldsymbol{\phi}$ and a parameter $\boldsymbol{\beta}$, uses the linear predictor $\mathbf{x}^\top\boldsymbol{\beta}$ and generates the target $y$ from this linear predictor by monotonically transforming it using a link function $\mathrm{g}_\phi = (\nabla\phi)^{-1}$ (see [3] for a table of GLM's and corresponding $\mathrm{g}_\phi$ or $\phi$). That is,

$$y = \mathrm{g}_\phi(\mathbf{x}) + \eta_\phi$$

where $\eta_\phi$ is a noise term from an exponential family distribution that depends on the GLM used (a detailed discussion on GLM's is available in the Supplement; see also [3] for examples of GLM's and their corresponding $\boldsymbol{\phi}$ and exponential family distribution).

**Loss Function: Bregman Divergences** Parameter estimation for GLM's involves minimisation of functions called Bregman Divergences, which are distance-like loss function that measure the "discrepancy" between predicted and true values in a GLM in terms of their log-likelihood. Corresponding Bregman Divergences for different GLMs are square loss for Gaussian, I-divergence for Poisson regression, etc. (see Table 1 in the Appendix). Given any two vectors $\mathbf{a}$ and $\mathbf{b}$, the Bregman divergence between them is defined as

$$D_{\boldsymbol{\phi}}(\mathbf{a}\|\mathbf{b}) \triangleq \boldsymbol{\phi}(\mathbf{a}) - \boldsymbol{\phi}(\mathbf{b}) - \langle\nabla\boldsymbol{\phi}(\mathbf{b}), \mathbf{a}-\mathbf{b}\rangle$$

We refer the reader to [3] for a detailed exposition on Bregman Divergences and their relationship with GLM's. For the purpose this manuscript, we use the fact that estimating the MLE parameter $\widehat{\boldsymbol{\beta}}$ from data $(\mathbf{X}, \mathbf{y})$ for a GLM parametrised by $\boldsymbol{\phi}$ is equivalent to the following optimisation problem:

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \sum_{(\mathbf{x},y)} D_{\boldsymbol{\phi}}\left(y\|\mathrm{g}_\phi(\mathbf{x}^\top\boldsymbol{\beta})\right) + \lambda\mathcal{R}(\boldsymbol{\beta})$$

For example, maximum likelihood for a Gaussian model/OLS corresponds to square loss, for Poisson it is the I-divergence, etc.

In particular, we note that the only aspect of our framework that is affected by generalising linear regression to GLMs is in the loss function, which now changes from a square loss to a general Bregman Divergence.

**3.2 Extension to Overlapping Aggregation** In section 2, we studied and derived an algorithm for the case when the aggregation is non-overlapping. We now generalise this to a wider class of aggregation paradigms. Consider the matrix $\boldsymbol{\Gamma} \in \mathbf{R}^{m\times n}$ such that $\boldsymbol{\Gamma}_{ki} = 1$ if and only if $i \in \mathcal{I}_k$, where $\mathcal{I}_k$ are the aggregation groupings as defined in section 2. Then, $\boldsymbol{\Gamma}\mathbf{y} = \mathbf{z}$ exactly captures the aggregate summaries $z_k = \sum_{i\in\mathcal{I}_k} y_i$.

Extensions to arbitrary aggregation paradigms is now equivalent to designing the appropriate $\boldsymbol{\Gamma}$ matrix. For example, overlapping aggregation can be represented by a $\boldsymbol{\Gamma}$ where multiple rows have 1's in the same column. A sliding window aggregation framework with window size $\tau$ can be represented with a $\boldsymbol{\Gamma}$ such that $\boldsymbol{\Gamma}_{ik} = 1$ for $i \in \{k\tau, k\tau+1, \cdots (k+1)\tau\}$, and 0 otherwise.

We can extend this further to the case where the aggregation is weighted– in this case, $\boldsymbol{\Gamma}$ is a matrix with general real-valued entries rather than a binary matrix, e.g. in Gaussian smoothing. This is common in applications involving non-uniformly sampled sub-populations, where aggregates need to computed from weighted averages in order to be an unbiased representation of the underlying population characteristics. Another example of weighted aggregation is when the target variables are released only after being obfuscated by multiplication with a random matrix in the interest of statistical disclosure limitation. All these cases can be handled in the same manner as for a binary matrix by using an appropriate $\boldsymbol{\Gamma}$ corresponding to the aggregation protocol.

**3.3 Learning Algorithm for the General Case** We are now ready to describe the modelling framework and the solution algorithm for the general case. Let $\boldsymbol{\phi}$ be the convex function on which the Bregman divergence corresponding to the GLM used is defined. We overload our notation and use $\mathrm{g}_\phi$ and $\mathrm{g}_\phi^{-1}$ to denote functions applied to the individual elements of the vector, that is for a vector $\mathbf{a}$, we have $\mathrm{g}_\phi^{-1}(\mathbf{a}) = [\mathrm{g}_\phi^{-1}(a_1), \mathrm{g}_\phi^{-1}(a_2)\cdots\mathrm{g}_\phi^{-1}(a_n)]$ whenever well defined. Let $\boldsymbol{\Gamma}$ be the aggregation matrix as defined in section 3.2. Then, the optimisation problem for the general version of the problem is

$$\begin{aligned} \min_{\mathbf{y},\boldsymbol{\beta}} \quad & D_{\boldsymbol{\phi}}\left(\mathbf{y} \| \mathrm{g}_{\phi}^{-1}(\mathbf{X}\boldsymbol{\beta})\right) + \lambda \mathcal{R}(\boldsymbol{\beta}) \\ \text{s.t.} \quad & \boldsymbol{\Gamma}\mathbf{y} = \mathbf{z} \end{aligned} \tag{3.3}$$

The steps involved in the overall algorithm for the general case (GLMs with arbitrary aggregation paradigms) is summarised as Algorithm 2. As earlier, we use alternating minimisation to solve this optimisation problem for the imputed targets $\mathbf{y}$ and parameter $\boldsymbol{\beta}$ respectively.

Given a particular set of values for the imputed targets $\mathbf{y}$, estimating the parameter $\boldsymbol{\beta}$ is a standard GLM parameter estimation step.

The more interesting problem here is the imputation of targets $\mathbf{y}$ given a particular value of $\boldsymbol{\beta}$, which can be summarised as:

$$\begin{aligned} \mathbf{y}^{+} = \quad \underset{\mathbf{y}}{\text{argmin}} \quad & D_{\boldsymbol{\phi}}\left(\mathbf{y} \| \mathrm{g}_{\phi}(\mathbf{X}\boldsymbol{\beta})\right) \\ \text{s.t.} \quad & \boldsymbol{\Gamma}\mathbf{y} = \mathbf{z} \end{aligned} \tag{3.4}$$

This is convex in $\mathbf{y}$, and in fact, the following result shows that we can obtain a closed form solution:

LEMMA 3.1. ***Optimal Target Imputation:*** *Given $\boldsymbol{\beta}$, the optimality conditions for the optimisation problem as described in equation 3.4 lead to the parameter $\mathbf{y}^{+}$, where $\mathbf{y}^{+}$ is defined as*

$$g_{\phi}\left[\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\Gamma}^{\top}\left(\boldsymbol{\Gamma}\boldsymbol{\Gamma}^{\top}\right)^{-1}\boldsymbol{\Gamma}\left(\mathbf{X}\boldsymbol{\beta} - g_{\phi}^{-1}\left((\boldsymbol{\Gamma}^{\top}\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}\mathbf{z}\right)\right)\right] \tag{3.5}$$

*where the operations $g_{\phi}$ and $g_{\phi}^{-1}$ are applied elementwise to their corresponding vector arguments.*

The proof of this follows directly by using either optimality conditions on the Lagrangian, or by using the Karush-Kuhn Tucker conditions for the constrained optimisation problem, we skip details for succinctness.

For better presentation and intuitive clarity, we separate out the target imputation step into three parts– they can, of course, be combined in any implementation as in equation 3.5. We use the generalised inverse in the matrix inversion steps whenever the matrices involved are not full rank.

**Convergence:** While the optimisation problem is not jointly convex, and the solution methodology in algorithms 1 and 2 involve alternating estimation and imputation steps, it can be shown that both the algorithms always converge.

LEMMA 3.2. ***Convergence:*** *For any choice of initialisation and any set of inputs, both algorithms 1 and 2 converge to a local minimum.*

**Algorithm 2** ESTIMAGG: PREMIUM
Arbitrary aggregation, Generalised Linear Models

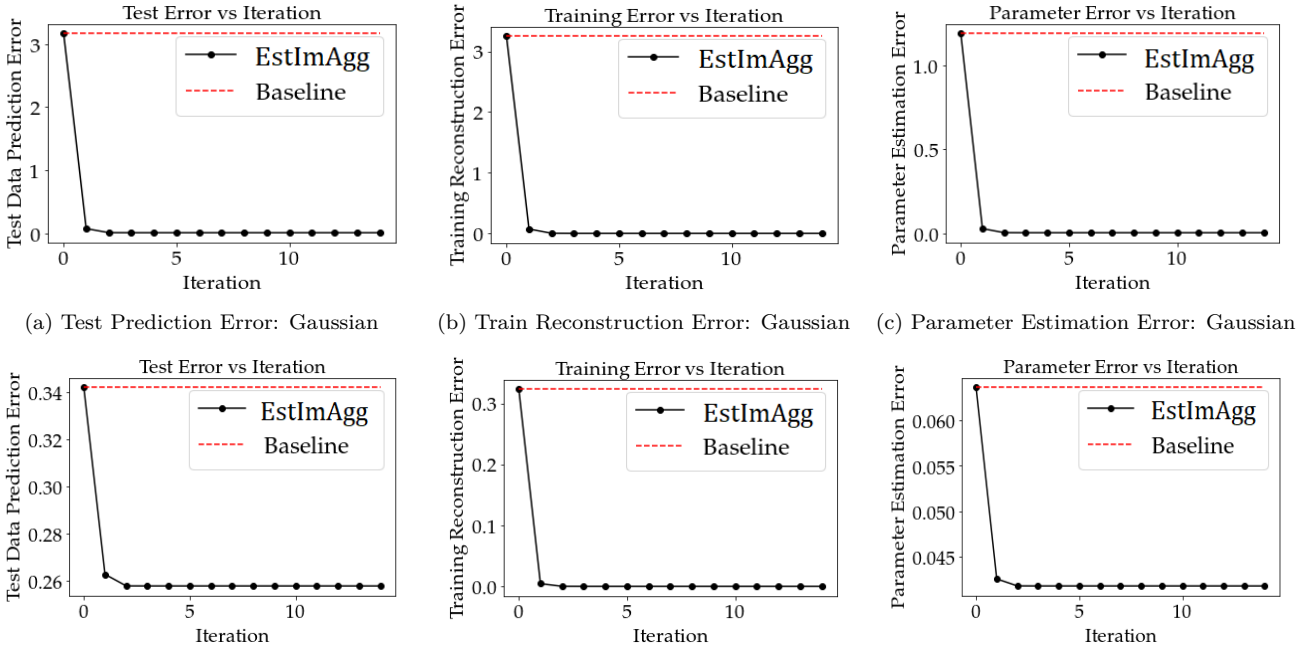
1: Input: $\mathbf{X}, \mathbf{z}$, GLM $\sim \boldsymbol{\phi}$, Aggregation matrix $\boldsymbol{\Gamma}$
2: Initialise $\{y_i = \sum_k \frac{z_k}{|\mathcal{I}_k|} : i \in \mathcal{I}_k\}$
3: **while** not converged **do**
4: Solve for $\boldsymbol{\beta}^{+}$ using standard GLM estimation
$$\boldsymbol{\beta}^{+} = \arg\min_{\boldsymbol{\beta}} D_{\boldsymbol{\phi}}\left(\mathbf{y} \| \mathrm{g}_{\phi}(\mathbf{X}\boldsymbol{\beta})\right) + \lambda \mathcal{R}(\boldsymbol{\beta})$$
5: Compute imputed aggregate error given new $\boldsymbol{\beta}^{+}$
$$\boldsymbol{\vartheta} = \left(\mathbf{X}\boldsymbol{\beta} - \mathrm{g}_{\phi}^{-1}\left((\boldsymbol{\Gamma}^{\top}\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}\mathbf{z}\right)\right)$$
6: Transform aggregate error
$$\boldsymbol{\xi} = \boldsymbol{\Gamma}^{\top}\left(\boldsymbol{\Gamma}\boldsymbol{\Gamma}^{\top}\right)^{-1}\boldsymbol{\Gamma}\boldsymbol{\vartheta}$$
7: Impute each target applying a monotonic transform
$$\mathbf{y}^{+} = \mathrm{g}_{\phi}\left(\mathbf{X}\boldsymbol{\beta} - \boldsymbol{\xi}\right)$$
8: Update variables $(\mathbf{y}, \boldsymbol{\beta}) = (\mathbf{y}^{+}, \boldsymbol{\beta}^{+})$
9: **end while**
10: **return** $\mathbf{y}, \boldsymbol{\beta}$

This fact can be proved using the observation that every step in the algorithm reduces the value of the objective function, and the fact that the objective function is bounded below by 0. The objective function is not jointly convex except for specific types of Bregman Divergences[1], hence convergence is local.

## 4 Experiments

We evaluate our methods on synthetic data as well as three real datasets with applications in online advertising and healthcare. As mentioned earlier, because of a lack of existing literature on this topic we are not aware of any algorithmic baselines for our work. The performance of our method, therefore, is compared against a straightforward ecological regression baseline [15] that treats aggregates $z$ as individual level samples and uses them for training.

The evaluation metrics that we use are threefold. First, following standard practice we examine the performance of our algorithms in predicting target values on an unseen test set compared to the true values. All of this is done at the individual-sample level of granularity. Next, note that while reconstruction is not an explicit objective, our algorithm nevertheless involves a data imputation step. Therefore, whenever we have access to ground truth data for the training set, we also evaluate the $\ell_2$ reconstruction error. Finally, whenever we have access to the "true" parameter of the GLM, we also compare the parameter recovery error.




(a) Test Prediction Error: Gaussian (b) Train Reconstruction Error: Gaussian (c) Parameter Estimation Error: Gaussian

(d) Test Prediction Error: Poisson (e) Train Reconstruction Error: Poisson (f) Parameter Estimation Error: Poisson

Figure 1: Synthetic Data: Gaussian (1a – 1c) and Poisson (1d – 1f): Our model outperforms the baseline in all metrics– test error, train reconstruction and parameter estimation– and converges within very few iterations

**4.1 Synthetic Data** We run two different experiments on synthetic data– one for a Gaussian regression model and one for a Poisson regression model. In both cases, we generate covariates $\mathbf{X}$ i.i.d Gaussian, as well as the true parameter $\boldsymbol{\beta}^*$ from a multivariate Gaussian. We then use these to generate the corresponding targets $\mathbf{y}$. We use $y|\mathbf{x} \sim \mathcal{N}(\mathbf{x}^\top \boldsymbol{\beta}^*, \sigma^2)$ for the Gaussian model. For the Poisson model, following standard practice [17, 3] we further apply the canonical exponential link function to generate each $y$ as $y|\mathbf{x} \sim \text{Poiss}\,(\eta_{\mathbf{x}})$, where $\eta_{\mathbf{x}} = exp(\mathbf{x}^\top \boldsymbol{\beta}^*)$.

In each case we generate the aggregation set $\boldsymbol{\Gamma}$ as a Bernoulli matrix $A_{ij} = Bernoulli(\rho)$ i.i.d. If $A_{ij} = 0$, we use $y_i$ to compute the $j^{th}$ aggregate, otherwise not. We use the aggregates as the "true" individual labels for our initialisation step (note that this is deterministic).

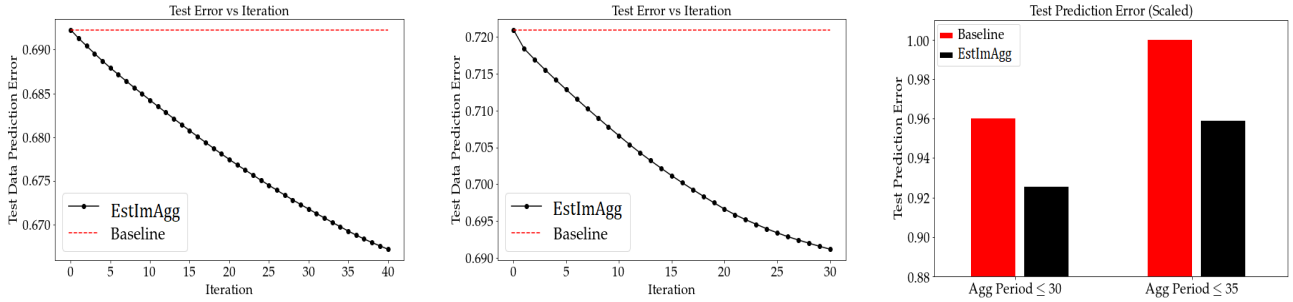
We then feed these aggregates to our algorithm together with the grouping information $\boldsymbol{\Gamma}$. The evaluation metrics are threefold. First, we compute prediction error at the individual level for each method on samples in the test. Next, we calculate error in reconstructing the training data as imputed by our method. Finally, since we have the "true" parameter for this set of experiments, we also plot parameter recovery error.

Figures 1 show the results for both the Gaussian and the Poisson regression model. In both cases, it can be seen that our algorithm significantly outperforms the baseline and converges within a very small number of iterations. The same experiments repeated with other values of $\rho$ and for varying problem size showed similar results, we omit the plots for lack of space.

**4.2 Real Data: CPC in Online Advertising** As described in Section 1, recall that while online advertising models need to predict individual-level CPC values in their bidding process, the training data for these models are usually only available in aggregated form (e.g.,the Google Shopping advertising product [21]). Our task here is to design effective predictive models for estimating CPC that can work at the per-click level but use only aggregates for training.

We evaluate our methods on a real-world proprietary aggregated-CPC dataset from Criteo [7], an online advertising company that provides personalised behaviourally retargeted advertising services for Internet retailers. We use a subset of Criteo's advertising data collected over a period of one week in February 2017. The dataset contains 25691 instances of ad-click data for different products, and each sample consists of the the CPC aggregate corresponding to its aggregate group, as well as a feature-set of size 1551 containing information about product-country, product-price, timestamp, campaign-type, etc. as well as taxonomic category information for each product in the dataset.

Since the number of times an advertisement gets clicked per day can be arbitrary, the aggregation period




(a) Test Error for Agg. over 30 clicks (b) Test Error for Agg. over 35 clicks (c) (Scaled) Test Error Comparison

Figure 2: Real Data: Estimating CPC for Online Advertising: Prediction error for test data plotted for aggregation over 30 clicks (figure 2a) and 35 clicks (figure 2b). Scaled errors shown in figure 2c for ease of comparison. Our model outperforms the baseline, leading to nearly 4-5% improvement in predictive performance

for each aggregate has been computed over varying numbers of clicks. Ideally, we would prefer to test the performance of our methods in predicting per-click CPC, but by design this information is not available for all data points. However, ground truth itemised CPC is nevertheless available for display ads that have only been clicked once during the entire aggregation period, hence in this case the aggregate is equal to the per-click CPC. Since the number of such samples are extremely limited, we use the entire set of single-click data points as test data, so that we can evaluate the performance of our predictive model at the granularity that is required for the real world application. The remaining data, which have only aggregated CPC information, is used for training.

Based on common industry practice in the ad-tech domain, we use a log-normal model [14] as our base GLM framework for predicting CPC. We perform our experiments on two different scenarios– first where the data has been aggregated over at most 30 clicks, and similarly for data aggregated over 35 clicks. Since ground truth information is not available for training data, we only show predictive performance on test data.

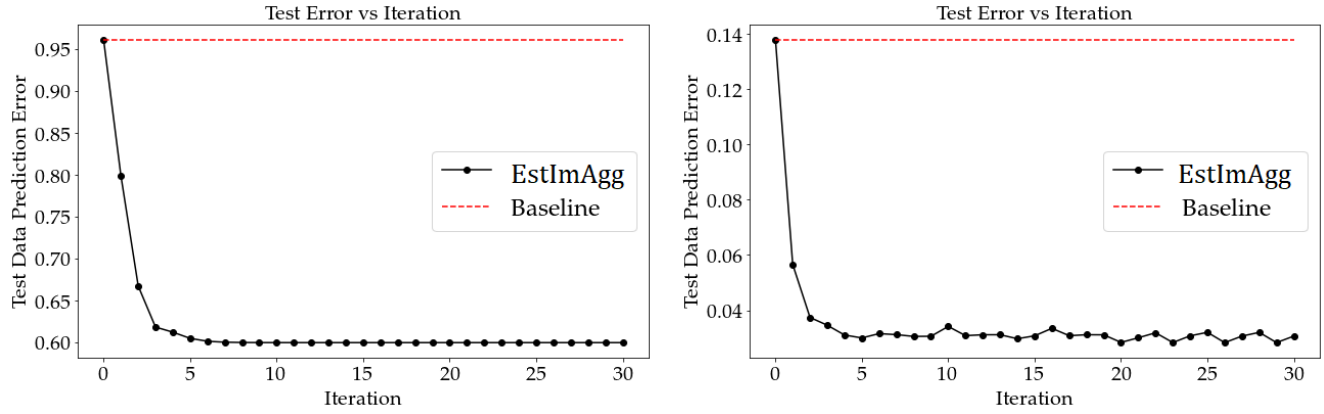
Figures 2 shows the plots of test error versus iteration for the two sets of experiments. For reasons of stability, we use a validation set to define a maximum number of iterations for the algorithm. In both cases, we can see that our algorithm results in a improvement in predictive performance over the baseline. For ease of understanding, we provide the scaled bar chart of the final average error values in figure 2c. It is clear that EstImAgg results in a nearly 4-5% improvement in estimation error. To put that in context, note that the online advertising industry sees billions of dollars in transactions per year[2] and even an improvement a few percentage points can indicate significant difference in revenue.

**4.3 Real Data: Healthcare** We presented our work so far within the context of predicting click-level CPC in the online advertising domain. However, aggregated data is common in many other fields, and our framework can be applied in an identical manner to domains beyond online advertising. Healthcare is one such domain where data aggregation arises naturally– privacy concerns regarding the confidentiality of patient information limits the kinds of data that can be released to the public, and statistical disclosure techniques like aggregation is one of the most popular techniques for this purpose. Patient information like healthcare charges, prevalence of pathological conditions, etc. are often released at the county or zip-code level, which can then be combined with publicly available census data to perform individual level predictive modelling.

We use two healthcare datasets to demonstrate the general applicability of our framework– the application involved is predicting Medicare reimbursements for the first dataset, and hospital charges for the second dataset.

Our first dataset is the CMS Beneficiary Summary DE-SynPUF dataset [8], which is a public use dataset created by the Centers for Medicare and Medicaid Services by applying different statistical disclosure limitation techniques to real beneficiary claims data in a way so as to very closely resemble real Medicare data. We use a subset of the DE-SynPUF dataset for Louisiana state from the year 2008 and model outpatient institutional annual primary payer reimbursement with all the available predictor variables that include age, race, sex, duration of coverage, presence of a variety of chronic conditions, etc. Because of the nature of our target variable, we use a Poisson regression model for this problem.

Our second dataset is the Texas Inpatient Discharge dataset from the TX Department of State Health Services [30]. Using hospital billing records from the fourth quarter of 2006, we regress patient expenditure on pre-




(a) Test Prediction Error on DESynPUF

(b) Test Prediction Error on TxID

Figure 3: Real Data: Test Error on DESynPUF (3a) and TXID dataset. Our model outperforms the baseline and converges within very few iterations, with a reasonably faithful reconstruction of the training data

dictor variables that include race, sex, county, zipcode, and length of stay (see [25] for more details on the dataset).

For both the datasets, we have ground truth information for our targets, which we use to form aggregates using Bernoulli random matrices in a manner identical to to what was done for synthetic data.

We evaluate the performance of our algorithm by computing the predictive error on the test set– the reconstruction error on the training data is deferred to the supplement for lack of space.

Figures 3a and 3b show the results for test data estimation error on DESynPUF and TxID respectively. For both these datasets, the plots show that our algorithm significantly outperforms the baseline and reaches a reasonably steady-state solution fairly rapidly within a few iterations. Results for varying granularities of aggregation were similar and are omitted for space constraints.

## 5 Discussion and Future Work

In this paper, we introduced a novel learning framework that can learn generalised linear models when the targets are only available as aggregates computed over arbitrary groupings of the data samples. This is an important learning paradigm in domains ranging from online advertising to healthcare where privacy and proprietary concerns limit the release of data at a granular level. We developed a new algorithm and empirically demonstrated its efficacy in learning under aggregation constraints.

It is instructive to examine the mathematical form predicated by the imputation step update for the targets– specifically, observe that the imputed targets for the linear model are obtained via an update in the form of a piecewise linear shift. This suggests a form of implicit "structural" regularisation– the model prefers imputed targets that adhere better to the linear relationship with the feature vectors.

It remains to be seen whether this can be used to design variations of this algorithm where the modelling framework or domain expertise suggests a different structural formalism– we defer further investigations in this direction to future work. Other potential directions worth pursuing would involve a rigorous statistical analysis of our algorithms including asymptotic consistency properties and sample complexity for finite sample error analyses.

## References

[1] S. Acharyya, O. Koyejo, and J. Ghosh. Learning to rank with bregman divergences and monotone retargeting. *arXiv preprint arXiv:1210.4851*, 2012.

[2] ADWEEK. U.S. Digital Advertising Will Make $ 83 Billion This YYear Says EMarketer, March 14, 2017. http://www.adweek.com/digital/u-s-digital-advertising-will-make-83-billion-this-yyear-says-emarketer/.

[3] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *The Journal of Machine Learning Research*, 6:1705–1749, 2005.

[4] A. Bhowmik, J. Ghosh, and O. Koyejo. Generalized Linear Models for Aggregated Data. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 93–101, 2015.

[5] A. Bhowmik, J. Ghosh, and O. Koyejo. Sparse parameter recovery from aggregated data. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1090–1099, 2016.

[6] A. Bhowmik, J. Ghosh, and O. Koyejo. Frequency domain

predictive modelling with aggregated data. In *Artificial Intelligence and Statistics*, pages 971–980, 2017.

[7] Criteo. https://www.criteo.com/.

[8] DESynPUF. 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.

[9] S. Dias, A. J. Sutton, A. Ades, and N. J. Welton. Evidence synthesis for decision making 2 a generalized linear modeling framework for pairwise and network meta-analysis of randomized controlled trials. *Medical Decision Making*, 33(5):607–617, 2013.

[10] J. J. Gart. The analysis of poisson regression with an application in virology. *Biometrika*, 51(3/4):pp. 517–521, 1964.

[11] A. Ghosh, B. I. Rubinstein, S. Vassilvitskii, and M. Zinkevich. Adaptive bidding for display advertising. In *Proceedings of the 18th international conference on World wide web*, pages 251–260. ACM, 2009.

[12] A. Goldfarb and C. Tucker. Online display advertising: Targeting and obtrusiveness. *Marketing Science*, 30(3):389–404, 2011.

[13] T. Jamil, W. A. Ozinga, M. Kleyer, and C. J. ter Braak. Selecting traits that explain species–environment relationships: a generalized linear mixed model approach. *Journal of Vegetation Science*, 24(6):988–1000, 2013.

[14] N. Johnson, S. Kotz, and N. Balakrishnan. Lognormal distributions. continuous univariate distributions (vol. 1), 1994.

[15] G. King, M. A. Tanner, and O. Rosen. *Ecological inference: New methodological strategies*. Cambridge University Press, 2004.

[16] A. C. Lozano, H. Li, A. Niculescu-Mizil, Y. Liu, C. Perlich, J. Hosking, and N. Abe. Spatial-temporal causal modeling for climate change attribution. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 587–596. ACM, 2009.

[17] P. McCullagh and J. A. Nelder. Generalized linear models. 1989.

[18] J. A. Nelder and R. Baker. *Generalized linear models*. Wiley Online Library, 1972.

[19] J. A. Nelder and R. W. M. Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society. Series A (General)*, 135(3):pp. 370–384, 1972.

[20] NORC. *General Social Survey*. http://www3.norc.org/GSS+Website/.

[21] NORC. *Google Shopping Ad Product*. https://developers.google.com/adwords/api/docs/appendix/reports/shopping-performance-report#averagecpc.

[22] U. D. of Commerce. *Bureau of Economic Analysis*. http://www.bea.gov/.

[23] U. D. of Labour. *Bureau of Labour Statistics*. http://www.bls.gov/.

[24] Y. Park and J. Ghosh. A probabilistic imputation framework for predictive analysis using variably aggregated, multi-source healthcare data. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 445–454. ACM, 2012.

[25] Y. Park and J. Ghosh. 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*, pages 55–64. ACM, 2014.

[26] G. Patrini, R. Nock, T. Caetano, and P. Rivera. (almost) no label no cry. In *Advances in Neural Information Processing Systems*, pages 190–198, 2014.

[27] C. Perlich, B. Dalessandro, T. Raeder, O. Stitelman, and F. Provost. Machine learning for targeted display advertising: Transfer learning in action. *Machine learning*, 95(1):103–127, 2014.

[28] N. Quadrianto, A. J. Smola, T. S. Caetano, and Q. V. Le. Estimating labels from label proportions. *The Journal of Machine Learning Research*, 10:2349–2374, 2009.

[29] W. S. Robinson. Ecological correlations and the behavior of individuals. *International journal of epidemiology*, 38(2):337–341, 2009.

[30] TxID. Texas Inpatient Public Use Data File. *Texas Department of State Health Services*, 2006. https://www.dshs.state.tx.us/thcic/hospitals/Inpatientpudf.shtm.

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

[32] J. Yan, N. Liu, G. Wang, W. Zhang, Y. Jiang, and Z. Chen. How much can behavioral targeting help online advertising? In *Proceedings of the 18th international conference on World wide web*, pages 261–270. ACM, 2009.

[33] Y. Yuan, F. Wang, J. Li, and R. Qin. A survey on real time bidding advertising. In *Service Operations and Logistics, and Informatics (SOLI), 2014 IEEE International Conference on*, pages 418–423. IEEE, 2014.

[34] R. L. Zeff and B. Aronson. *Advertising on the Internet*. John Wiley & Sons, Inc., 1999.