Training Data Subset Selection for Regression With Controlled Generalization Error

Durga Sivasubramanian [1] **Rishabh Iyer** [2] **Ganesh Ramakrishnan** [1] **Abir De** [1]

Abstract

Data subset selection from a large number of training instances has been a successful approach toward efficient and cost-effective machine learning. However, models trained on a smaller subset may show poor generalization ability. In this paper, our goal is to design an algorithm for selecting a subset of the training data, so that the model can be trained quickly, without significantly sacrificing on accuracy. More specifically, we focus on data subset selection for L_2 regularized regression problems and provide a novel problem formulation which seeks to minimize the training loss with respect to both the trainable parameters and the subset of training data, subject to error bounds on the validation set. We tackle this problem using several technical innovations. First, we represent this problem with simplified constraints using the dual of the original training problem and show that the objective of this new representation is a monotone and α-submodular function, for a wide variety of modeling choices. Such properties lead us to develop SELCON, an efficient majorization-minimization algorithm for data subset selection, that admits an approximation guarantee even when the training provides an imperfect estimate of the trained model. Finally, our experiments on several datasets show that SELCON trades off accuracy and efficiency more effectively than the current state-of-the-art.

1. Introduction

Data-driven estimation of the underlying statistical model is the central challenge in any supervised machine learning (ML) problem. Thanks to the law of large numbers (Casella & Berger, 2002), such a training procedure often demands a huge number of training examples to ensure statistical reliability of the learned model. Therefore, the success of several machine learning models can be attributed to the availability of a massive amount of data and thus to the high performance computing infrastructures, *e.g.*, GPUs, multicore processors, high storage disks, *etc.*, which are required to store and process such data. These computational resources involve large expenses, additional energy utilization and maintenance costs. Mitigation of such overheads without sacrificing the accuracy of the predictive model is a challenging task, which often entails a careful selection of a smaller number of training instances, so that the training algorithm can be run in an environment with limited resources (Lucic et al., 2017; Mirzasoleiman et al., 2020; Boutsidis et al., 2013; Kaushal et al., 2019; Killamsetty et al., 2021b; Wei et al., 2014a; Liu et al., 2015; Bairi et al., 2015; Kirchhoff & Bilmes, 2014). However, current data selection techniques do not explicitly account for the generalization error which may be exacerbated in the presence of a small sized training dataset. As a consequence, they can suffer from high generalization error, especially for large datasets.

1.1. Present work

In response to the above limitations, our goal is to select a subset from training data in such a way that the model can be quickly trained in an environment with limited resources, while at the same time, provide good predictive power. More specifically, we make the following contributions.

Novel formulation of data selection. In this work, we focus on the regression problem and introduce a novel problem formulation (Section 2) - which encodes the task of data selection for regression, while ensuring that the error on validation set remains below an acceptable level. Such an explicit use of the validation set during training improves the generalization ability of the inferred model, as indicated in (Ren et al., 2018; Killamsetty et al., 2021b).

More specifically, given a model class and a fixed validation set, we seek to minimize an L_2 regularized constrained squared error loss with respect to both the parameter vector and the subset of training data, subject to a set of error bounds on different portions of the validation set. The use of such error bounds as optimization constraints enhances the generalization ability of the inferred model in the face

[1]CSE Department, Indian Institute of Technology, Bombay [2]CS Department, University of Texas at Dallas. Correspondence to: Durga Sivasubramanian <durgas@cse.iitb.ac.in>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

of small training data. Moreover, the presence of multiple error constraints in our setup can be useful in several data selection problems; *e.g.*, learning with heterogeneous data where each constraint limits the error for each cluster of data (Rothenhäusler et al., 2018); fair regression with bounded group loss, where each constraint limits the error on the protected group(s), *etc.* (Agarwal et al., 2019).

In general, our data selection problem is NP-hard due to the presence of both the training set and the model parameters as optimization variables. However, it permits us to reformulate it into a new optimization task with simplified constraints, by making use of the Lagrangian dual of the original training problem. This new optimization problem can be seen as an instance of cardinality-constrained set function minimization problem, where the objective corresponds to the optimal training loss as a function of the candidate data subset.

Characterizing the loss function of data selection. Having represented the optimal training loss as a set function, we show that this function is monotone, α-submodular (Gatmiry & Gomez-Rodriguez, 2018; Lehmann et al., 2006; Hassani et al., 2017) and enjoys a bounded generalized curvature (Iyer et al., 2013b; Zhang & Vorobeychik, 2016) for a wide variety of models including a class of nonlinear functions (Section 3). These technical results can be useful in other related data selection problems and therefore, are of independent interest.

Approximation algorithm for data selection. Finally, to solve our data selection problem, we design SELCON, a new majorization-minorization algorithm (Algorithm 1, Section 4) building upon the semi-differentials proposed by (Iyer et al., 2013a; Iyer & Bilmes, 2015), which minimizes the set function characterized above. SELCON enjoys an approximation guarantee even when the training algorithm provides an imperfect estimate. While doing so, we obtain a new family of modular upper bounds of an α-submodular function, which extends the bounds proposed in (Iyer et al., 2013a) and therefore, can be of independent technical interests. Moreover, SELCON can minimize any monotone, α-submodular function, going beyond the particular instance in this work, which makes it useful from a broader perspective.

We evaluate[1] our framework on several real-world datasets, and demonstrate that SELCON trades off the accuracy and efficiency more effectively than several baselines and state-of-the-art. We also demonstrate that the use of constrained validation set error maintains the generalization ability of the inferred model in the presence of small training data. Finally, we test our framework on the application of fair regression with bounded group loss, which shows that SELCON offers fair prediction along with an effective trade-off between accuracy and efficiency.

[1]Our code and data is available at https://github.com/abir-de/SELCON

1.2. Related work

Algorithms for data selection predominantly follow two approaches. The first approach (Wei et al., 2014a;b; Liu et al., 2015; Bairi et al., 2015) selects diverse training examples by maximizing submodular *proxy* functions, *e.g.*, facility location, *etc.*, and *then* use them to train the underlying model. The second approach selects *coresets* – weighted subsets of training examples – alongside training the model over them (Lucic et al., 2017; Mirzasoleiman et al., 2020; Killamsetty et al., 2021a; Campbell & Broderick, 2018; Boutsidis et al., 2013; Kaushal et al., 2019). The choice of a coreset depends strongly on the model as well as on the training loss. Therefore, coreset selection algorithms vary widely across different ML settings, *e.g.*, SVM (Clarkson, 2010), Bayesian inference (Campbell & Broderick, 2018), k-means clustering (Har-Peled & Mazumdar, 2004), regression (Boutsidis et al., 2013), deep learning (Mirzasoleiman et al., 2020; Killamsetty et al., 2021a), *etc.*However, they do not explicitly control the validation set error, which often constrains their predictive power.

Our work is related to robust and efficient learning methods (Ren et al., 2018; Zhang & Sabuncu, 2018; Killamsetty et al., 2021b), that utilize the validation set to improve the training performance via a bi-level optimization. However, these approaches do not explicitly control the validation set error the way we do. Our work is also related to subset selection problems in the context of human-assisted machine learning (De et al., 2020; 2021), that aim to select a training subset to outsource to humans, rather than facilitating efficient learning. Moreover, unlike us, these setups do not consider any validation constraint. Our work is also connected with batch active learning methods (Wei et al., 2015; Hashemi et al., 2019; Kulkarni et al., 2018; Sener & Savarese, 2018), that aim to select examples from training data in order to minimize the labeling cost. In contrast, our setup has access to all the labels and it aims to select data to improve efficiency.

In recent years, there is a flurry of works on maximizing non-submodular functions (Horel & Singer, 2016; Das & Kempe, 2011; Bian et al., 2017; Kuhnle et al., 2018; Gatmiry & Gomez-Rodriguez, 2018; Hassidim & Singer, 2018; 2017). However, there is a paucity of work on minimizing α-submodular functions. Very recently, El Halabi & Jegelka (2020) aim to minimize the difference between two monotone α-submodular and β-submodular functions. However, they do not consider a cardinality constraint, which makes their approach less relevant to our setting.

2. Problem formulation

In the following, we first setup the notation and contextualize our problem. Thereafter, we formally present our data selection problem which involves simultaneous selection of a subset $\mathcal{S}$ of the training dataset $\mathcal{D}$ and training of a regression model $y \approx h_{\boldsymbol{w}}(\boldsymbol{x})$, subject to validation error constraints. We obtain an alternative representation of this problem, using the Lagrangian dual of the parameter estimation task. Finally we formally show that our data selection problem is NP-Hard.

2.1. Notation

Let $\{\boldsymbol{x}_i, y_i\}_{i\in\mathcal{D}}$ be the set of training samples and $\{\boldsymbol{x}_j, y_j\}_{j\in\mathcal{V}}$ the set of validation samples. Here, $\boldsymbol{x}_\bullet \in \mathbb{R}^d$ are the features and $y_\bullet \in \mathbb{R}$ are the corresponding response (output) variables. We also have a partition of Q subsets over the validation set, *i.e.*, $\mathcal{V} = V_1 \cup V_2 \cup \ldots \cup V_Q$. Unless otherwise stated, $\|\cdot\|$ denotes the L_2 norm, *i.e.*, $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$.

2.2. Our broad objective

We are provided a modeling framework $h_{\boldsymbol{w}} : \mathbb{R}^d \to \mathbb{R}$ which can approximate the relationship between $\boldsymbol{x}$ and y, *i.e.*, $y \approx h_{\boldsymbol{w}}(\boldsymbol{x})$, where $\boldsymbol{w}$ is a trainable parameter vector. Given the aforementioned setup, one can learn $\boldsymbol{w}$ using standard least square estimation. In principle, one might be tempted to estimate $\boldsymbol{w}$ using the entire set of training examples $\mathcal{D}$, which would possibly give a statistically sound estimate of $\boldsymbol{w}$. However, if the size of $\mathcal{D}$ is large, such exhaustive training may be inefficient in a typical computing environment. To tackle this problem, our goal is to determine a smaller subset of training samples $\mathcal{S} \subset \mathcal{D}$ such that it allows for efficient training of the model without significant drop in accuracy.

2.3. Problem setting for data selection

Given the full training set $\{\boldsymbol{x}_i, y_i\}_{i\in\mathcal{D}}$ and the validation set $\{\boldsymbol{x}_j, y_j\}_{j\in\mathcal{V}}$ along with its partitions $\mathcal{V} = \cup_{q\in[Q]} V_q$ and the model class $h_{\boldsymbol{w}}$, we consider minimization of the L_2 regularized training loss, jointly with respect to parameters $\boldsymbol{w}$ and the candidate subset $\mathcal{S}$, subject to a set of constraints that bound the mean squared errors (MSE) on the Q partitioning subsets of the validation set, *i.e.*,

$$\begin{aligned} &\underset{\mathcal{S}\subset\mathcal{D},\boldsymbol{w}}{\text{minimize}} \quad \sum_{i\in\mathcal{S}} [\lambda \|\boldsymbol{w}\|^2 + (y_i - h_{\boldsymbol{w}}(\boldsymbol{x}_i))^2], \\ &\text{subject to,} \quad \frac{\sum_{j\in V_q}(y_j - h_{\boldsymbol{w}}(\boldsymbol{x}_j))^2}{|V_q|} \le \delta, \ \forall q \in [Q], \\ &\qquad\qquad |\mathcal{S}| = k. \end{aligned} \tag{1}$$

Here, λ is the coefficient of the regularizer; the cardinality constraint limits the number of training samples to be chosen; and the validation error constraints ensure that the predictor's loss remains below some acceptable level δ for the subsets $\{V_q\}$ of the validation set[2].

Discussion on multiple validation error bounds. Note that the absence of validation error constraints in the basic problem setting may result in efficient training, but might not generalize well owing to small size of the training data. The validation error constraints in (1) ameliorate this problem, by attenuating the generalization error which might have exacerbated in the face of a small sized training data.

We note that, in order to improve the generalization ability, one may consider bounding the MSE on the entire validation set as one single constraint, *viz.*, $\frac{1}{|\mathcal{V}|}\sum_{j\in\mathcal{V}}(y_j - h_{\boldsymbol{w}}(\boldsymbol{x}_j))^2 \le \delta$, rather than constraining the MSE for multiple subsets of the validation set as in Eq. (1). However, we envision the use of formulation (1) in several applications. For example, in the case of fair regression with bounded group loss, the validation set can be partitioned in a way that each subset V_q corresponds to the sub-population for a protected group, so that the individual MSE for each protected group remains small. Our setup can also be useful in learning from heterogeneous data, wherein the heterogeneity could have arisen owing to multiple sources of data, time-shifts in the distribution, *etc.*. To address such requirements, the validation set can be partitioned into different subsets, where each subset represents a partition with similar properties.

2.4. A soft-constraint approach

It is evident that arbitrarily reducing δ would eventually make the error constraints infeasible in the above optimization problem (1). Therefore, we relax the constraints by provisioning for some margin of violation of these constraints. To this aim, we introduce new slack variables $\xi_1, \xi_2, ..., \xi_Q$ and replace each hard validation error (inequality) constraint in Eq. (1) by a soft constraint, *i.e.*, $\frac{1}{|V_q|}(y_j - h_{\boldsymbol{w}}(\boldsymbol{x}_j))^2 \le \delta + \xi_q$ similar to the soft-SVM formulation. Here ξ_q measures the extent of error violation in the constraint $\frac{1}{|V_q|}(y_j - h_{\boldsymbol{w}}(\boldsymbol{x}_j))^2 \le \delta$. Finally, we minimize the sum of regularized loss computed over the candidate set $\mathcal{S}$, along with a weighted sum of the slack variables that penalizes the constraint violation to yield the optimization problem in Eq. (2), *i.e.*,

$$\begin{aligned} &\underset{\mathcal{S}\subset\mathcal{D},\boldsymbol{w},\{\xi_q\}_{q\in[Q]}}{\text{minimize}} \quad \sum_{i\in\mathcal{S}} [\lambda \|\boldsymbol{w}\|^2 + (y_i - h_{\boldsymbol{w}}(\boldsymbol{x}_i))^2] + C\sum_{q\in V_q} \xi_q, \\ &\text{such that,} \quad \frac{\sum_{j\in V_q}(y_j - h_{\boldsymbol{w}}(\boldsymbol{x}_j))^2}{|V_q|} \le \delta + \xi_q \quad \forall q \in [Q], \\ &\qquad\qquad \xi_q \ge 0 \quad \forall\, q \in [Q] \text{ and, } |\mathcal{S}| = k \end{aligned} \tag{2}$$

where $\{\xi_q\}$ are the optimization variables in addition to the parameter vectors $\boldsymbol{w}$ and the candidate set $\mathcal{S}$ that were

[2] For the sake of brevity, we assumed the same value of δ across different validation subsets $\{V_q\}$.

already specified in (1). Through C, we can control the extent of penalization on the of violation of the validation set error. We note that as $C \to \infty$, the above formulation becomes equivalent to its hard constrained counterpart (1).

We may consider two possible approaches to solve the optimization problem in Eq. (2). In the first approach, we initially minimize the optimization problem (2) with respect to $\mathcal{S}$ for fixed $\boldsymbol{w}$ and $\{\xi_q\}$; and thereafter minimize the inner optimization objective with respect to $\boldsymbol{w}$ and $\{\xi_q\}$. This can be viewed as an instance of minimizing the sum of k smallest elements, which we expect to be intractable, since it is a concave minimization problem. In the second approach, given a fixed set $\mathcal{S}$ we first minimize (2) with respect to $\boldsymbol{w}$ and $\{\xi_q\}$; and thereafter, minimize this quantity with respect to $\mathcal{S}$. In this work, we focus on the second approach, which, as we will show in Section 4, provides a tractable solution with an approximation guarantee.

For any given set $\mathcal{S}$, let the optimal value of the parameters be $\boldsymbol{w}^*(\mathcal{S})$ and $\xi_q^*(\mathcal{S})$. We note that, if we define,

$$g(\mathcal{S}) = \sum_{i\in\mathcal{S}} [\lambda \left\|\boldsymbol{w}^*(\mathcal{S})\right\|^2 + (y_i - h_{\boldsymbol{w}^*(\mathcal{S})}(\boldsymbol{x}_i))^2] + C \sum_{q\in[Q]} \xi_q^*(\mathcal{S}), \tag{3}$$

then, our data selection problem becomes equivalent to

$$\underset{\mathcal{S}}{\text{minimize}}\, g(\mathcal{S}), \quad \text{subject to, } |\mathcal{S}| = k. \tag{4}$$

2.5. Representation of Eq. (2) with simplified constraints

Next, we obtain an alternative representation of the data selection problem, by making use of the Lagrangian dual[3] of the optimization problem (2) for a fixed $\mathcal{S}$, as formalized in the following proposition (Proven in Appendix A.1 in the supplementary material). As we shall discuss, such a new representation becomes equivalent to Eq. (4) for convex loss functions.

Proposition 1 *Given a fixed training set $\mathcal{S}$, let $\boldsymbol{\mu} = [\mu_q]_{q\in[Q]}$ be the Lagrangian multipliers for the constraints $\{\frac{1}{|V_q|}\sum_{j\in V_q}(y_j - h_{\boldsymbol{w}}(\boldsymbol{x}_j))^2 \le \delta + \xi_q\}_{q\in[Q]}$ in the optimization problem (2) and $F(\boldsymbol{w}, \boldsymbol{\mu}, \mathcal{S})$ be defined as follows:*

$$F(\boldsymbol{w}, \boldsymbol{\mu}, \mathcal{S}) = \sum_{i\in\mathcal{S}} [\lambda \left\|\boldsymbol{w}\right\|^2 + (y_i - h_{\boldsymbol{w}}(\boldsymbol{x}_i))^2] + \sum_{q\in[Q]} \mu_q \left[\frac{\sum_{j\in V_q}(y_j - h_{\boldsymbol{w}}(\boldsymbol{x}_j))^2}{|V_q|} - \delta\right] \tag{5}$$

Then, for the fixed set $\mathcal{S}$, the dual of the optimization problem (2) for estimating $\boldsymbol{w}$ and $\{\xi_q\}$ is given by,

$$\underset{\mathbf{0}\le\boldsymbol{\mu}\le C\mathbf{1}}{\text{maximize}}\ \underset{\boldsymbol{w}}{\text{minimize}}\ F(\boldsymbol{w}, \boldsymbol{\mu}, \mathcal{S}) \tag{6}$$

[3]The dual is formed with respect to the model parameters $\boldsymbol{w}$ and $\{\xi_q\}$, which allows us to augment the validation error constraints and $\{\xi_q\} \ge 0$ into the new objective. However, it still remains as a constrained optimization problem with respect to $\mathcal{S}$.

Let the inner minimization sub-problem of the above optimization problem have the solution $\boldsymbol{w}^*(\boldsymbol{\mu}, \mathcal{S})$ for a given $\boldsymbol{\mu}$ and $\mathcal{S}$. If the corresponding outer maximization problem has the solution $\boldsymbol{\mu}^* = \boldsymbol{\mu}^*(\mathcal{S})$ for a given $\mathcal{S}$, then the above dual problem has an optimal solution at $(\boldsymbol{w}^*(\boldsymbol{\mu}^*(\mathcal{S}), \mathcal{S}), \boldsymbol{\mu}^*(\mathcal{S}))$. To this end, given any set $\mathcal{S}$, we write the solution of this dual problem as the following set function.

$$f(\mathcal{S}) = F(\boldsymbol{w}^*(\boldsymbol{\mu}^*(\mathcal{S}), \mathcal{S}), \boldsymbol{\mu}^*(\mathcal{S}), \mathcal{S}) \tag{7}$$

Subsequently, we aim to select $|\mathcal{S}|$ by solving the following optimization problem.

$$\underset{\mathcal{S}\subset\mathcal{D}}{\text{minimize}}\, f(\mathcal{S}) \qquad \text{such that, } |\mathcal{S}| = k. \tag{8}$$

Relation between $f(\mathcal{S})$ and $g(\mathcal{S})$. Given a fixed $\mathcal{S}$, the optimization problems (2) and (6) are equivalent for convex losses. However, they may not be equivalent for non-convex losses and, by weak-duality, $f(\mathcal{S})$ would serve as a lower bound for $g(\mathcal{S})$. This leads us to the following proposition.

Proposition 2 *Given that $f(\cdot)$ and $g(\cdot)$ are defined in Eqs. (3) and (7) respectively, $f(\mathcal{S}) \le g(\mathcal{S})$ and the equality holds if the loss $(y - h_{\boldsymbol{w}}(\boldsymbol{x}))^2$ is convex with respect to $\boldsymbol{w}$. Hence, $\min_{\mathcal{S},|\mathcal{S}|=k} f(\mathcal{S}) \le \min_{\mathcal{S},|\mathcal{S}|=k} g(\mathcal{S})$.*

2.6. Differences with weighted sum of training and validation loss

Weighted sum of training and validation loss. Instead of our model, one can consider minimizing a weighted combination of training and validation losses, as follows:

$$\underset{\boldsymbol{w},\eta}{\text{minimize}}\ \eta \sum_{i\in\mathcal{S}} \left[\lambda||\boldsymbol{w}||^2 + (y_i - h_{\boldsymbol{w}}(\boldsymbol{x}_i))^2\right] + (1-\eta)\ (k/|V|) \sum_{j\in V} (y_j - h_{\boldsymbol{w}}(\boldsymbol{x}_j))^2 \tag{9}$$

The multiplier $k/|V|$ in the second term above ensures correct scaling w.r.t. the first term. Now, along with η can be estimated in two ways.

η is a hyperparameter: We can treat η as hyperparameter and cross validate them on another validation set V'. However, due to the requirement for tuning this additional hyperparameter, this approach is extremely time consuming and therefore, is not suitable for efficient data selection.

η is a trainable parameter: In this alternative approach, we train the η along with $\boldsymbol{w}$. Such a setup uses no additional validation set V'. However, since $\min_{\eta\in[0,1]}(a\eta + (1-\eta)b) = \min\{a, b\}$, the problem (9) reduces to

$$\underset{\mathcal{S},\boldsymbol{w}}{\text{minimize}}\ \min\left\{\sum_{i\in\mathcal{S}} \left[\lambda||\boldsymbol{w}||^2 + (y_i - h_{\boldsymbol{w}}(\boldsymbol{x}_i))^2\right], \frac{k}{|V|}\sum_{j\in V_q} (y_j - h_{\boldsymbol{w}}(\boldsymbol{x}_j))^2\right\} \tag{10}$$

Hence, it can latch on either minimizing only training set error *or* only validation set error, which results in (i) train-

ing only on validation set *or,* (ii) selecting subset without controlling generalization error.

Our approach. In our work, the Lagrangian multipliers $\boldsymbol{\mu}$ of the dual objective $F(\boldsymbol{w}, \boldsymbol{\mu}, \mathcal{S})$ defined in Eq. (5) can also be viewed as weights for validation error. However, we neither treat them as hyperparameters, nor learn them by simply minimizing the objective as in Eq. (9) above. Rather, our formulation naturally casts a max-min optimization task described in Eq. (6), that trains $\boldsymbol{w}$ and $\boldsymbol{\mu}$ in an adversarial manner. This also ensures that the validation error is not much higher than δ. In contrast, the formulation in Eq. (9) neither trains $\boldsymbol{w}$ and η using max-min optimization nor incorporates δ.

2.7. Hardness analysis for our approach

Given any fixed training subset $\mathcal{S}$, we can learn the optimal solution of the problem (2) using a standard optimization technique. In fact, it can be computed in polynomial time if the loss $(y - h_{\boldsymbol{w}}(\boldsymbol{x}))^2$ is convex. However, simultaneously determining the optimal set $\mathcal{S}^*$ and the optimal parameters $\boldsymbol{w}^*$ for that optimal set is not possible in polynomial time, as suggested by the following proposition (proof in Appendix A.2 in the supplementary).

Proposition 3 *Both the variants of the data selection problems* (4) *and* (8) *are NP-Hard.*

We will focus on minimizing $f(\mathcal{S})$ rather than $g(\mathcal{S})$, since that allows us to design a tractable algorithm with approximation guarantee for a wide class of modeling choices including nonlinear functions, and which works well in practice. Moreover, since $f(\mathcal{S}) = g(\mathcal{S})$ for convex losses, such an approximation guarantee also holds for $g(\mathcal{S})$ in the specific case of linear regression.

3. Characterization of $f(\mathcal{S})$

We next show that $f(\mathcal{S})$ is monotone and α-submodular and then, bound its generalized curvature, which would be subsequently used to design an efficient approximation algorithm for the optimization problem in Eq. (8). To help formally state the results, we begin with defining the following properties.

Definition 4 *Given a ground set $\mathcal{D}$ and a set function $f : 2^{\mathcal{D}} \to \mathbb{R}$, let us define $f(a \,|\, \mathcal{S}) = f(\mathcal{S} \cup \{a\}) - f(\mathcal{S})$. Then we have the following definitions.*

1. Monotonicity: *$f(\cdot)$ is monotone if $f(a \,|\, \mathcal{S}) \geq 0$ for all $\mathcal{S} \subset \mathcal{D}$ and $a \in \mathcal{D} \backslash \mathcal{S}$.*
2. α-submodularity: *$f(\cdot)$ is α-submodular with the submodularity parameter $\alpha > 0$, if for $\mathcal{S} \subseteq \mathcal{T}$ and $a \in \mathcal{D} \backslash \mathcal{T}$, we have $f(a \,|\, \mathcal{S}) \geq \alpha\, f(a \,|\, \mathcal{T})$ (Hashemi et al., 2019; Zhang & Vorobeychik, 2016; El Halabi & Jegelka, 2020).*
3. Generalized curvature: *Given a set $\mathcal{S}$, the generalized curvature of $f(\mathcal{S})$ is defined as (Iyer et al., 2013b; Zhang & Vorobeychik, 2016)*

$$\kappa_f(\mathcal{S}) = 1 - \min_{a \in \mathcal{D}} \frac{f(a|\mathcal{S}\backslash\{a\})}{f(a|\emptyset)}. \tag{11}$$

Note that, α-submodularity is a natural extension of submodularity. An α-submodular function $f(\mathcal{S})$ is submodular if $\alpha = 1$. Moreover, note that an $\kappa_f(\mathcal{S}) \geq 1 - 1/\alpha$. For a general monotone function f, $\alpha \leq 1$.

3.1. Monotonicity of $f(\mathcal{S})$

We formalize the monotonicity of $f(\mathcal{S})$, as defined in Eq. (7), in the following proposition (proof in Appendix B.1).

Proposition 5 *For any model $h_{\boldsymbol{w}}$, $f(\mathcal{S})$ is monotone,* i.e., *$f(\mathcal{S} \cup \{a\}) - f(\mathcal{S}) \geq 0$ for all $\mathcal{S} \subset \mathcal{D}$ and $a \in \mathcal{D} \backslash \mathcal{S}$.*

3.2. α-submodularity of $f(\mathcal{S})$

Next, we set about to present our key results on α-submodularity of $f(\mathcal{S})$ for different modeling choices of $h_{\boldsymbol{w}}$. To this aim, we first characterize the submodularity parameter of $f(\mathcal{S})$ for any bounded Hessian nonlinear model, in terms of λ, C, δ and some specific properties of the dataset (proof in Appendix B.2).

Theorem 6 *Assume that $|y| \leq y_{\max}$; $h_{\boldsymbol{w}}(\boldsymbol{x}) = 0$ if $\boldsymbol{w} = \mathbf{0}$, i.e., $h_{\boldsymbol{w}}(\boldsymbol{x})$ has no bias term; $h_{\boldsymbol{w}}$ is H-Lipschitz, i.e., $|h_{\boldsymbol{w}}(\boldsymbol{x})| \leq H \|\boldsymbol{w}\|$; the eigenvalues of the Hessian matrix of $(y - h_{\boldsymbol{w}}(\boldsymbol{x}))^2)$ have a finite upper bound, i.e., $\text{Eigenvalue}(\nabla^2_{\boldsymbol{w}}(y - h_{\boldsymbol{w}}(\boldsymbol{x}))^2) \leq 2\chi^2_{\max}$; and, define $\ell^* = \min_{a \in \mathcal{D}} \min_{\boldsymbol{w}} \ \chi^2_{\max} \cdot \|\boldsymbol{w}\|^2 + (y_a - h_{\boldsymbol{w}}(\boldsymbol{x}_a))^2 > 0$. Then, for $\lambda \geq \max\left\{\chi^2_{\max}, 32(1+CQ)^2 y^2_{\max} H^2/\ell^*\right\}$, $f(\mathcal{S})$ is a α-submodular set function, where*

$$\alpha \geq \widehat{\alpha}_f = 1 - \frac{32(1+CQ)^2 y^2_{\max} H^2}{\lambda \ell^*}, \tag{12}$$

Note that as $\lambda \to \infty$, we have $\alpha \to 1$, which implies that for large λ, $f(\mathcal{S})$ becomes close to submodular.

Proof sketch: The proof of the above theorem consists of two steps. In the first step, we show that $f(\mathcal{S} \cup \{a\}) - f(\mathcal{S}) \geq \min_{\boldsymbol{w}} \lambda ||\boldsymbol{w}||^2 + (y_i - h_{\boldsymbol{w}}(\boldsymbol{x}_a))^2$. Next, we derive that $f(\mathcal{T} \cup \{a\}) - f(\mathcal{T}) \leq \lambda \left\|\boldsymbol{w}^*(\boldsymbol{\mu}^*(\mathcal{T} \cup a), \mathcal{T})\right\|^2 + (y_a - h_{\boldsymbol{w}^*(\boldsymbol{\mu}^*(\mathcal{T} \cup a), \mathcal{T})}(\boldsymbol{x}_a))^2$. Finally, we use different properties of $f(\cdot)$ and the data to get a lower bound on the ratio of the above two quantities.

For a linear model $h_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$, we exploit additional properties of the underlying model to obtain a slightly tighter bound (Proven in Appendix B.3).

Proposition 7 *Given $0 < y_{\min} \leq |y| \leq y_{\max}$, $h_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$, $\|\boldsymbol{x}\| \leq x_{\max}$, we set the regularizing coefficient as $\lambda \geq \max\left\{x^2_{\max}, 16(1+CQ)^2 y^2_{\max} x^2_{\max}/y^2_{\min}.\right\}$. Then*

$f(\mathcal{S})$ is a α-submodular set function, where

$$\alpha \geq \widehat{\alpha}_f = 1 - \frac{16(1+CQ)^2 y_{\max}^2 x_{\max}^2}{\lambda y_{\min}^2}. \quad (13)$$

Subset selection for linear regression problems has been widely studied in literature (Hashemi et al., 2019; De et al., 2020). Most often, these approaches optimize measures associated with the covariance matrix, rather than explicitly minimizing the training loss subject to the validation set error bound.

3.3. Generalized curvature

Next, we provide a unified bound on the generalized curvature (*c.f.*, Definition 4) for both linear and nonlinear modeling choices of $h_{\boldsymbol{w}}(\boldsymbol{x})$, as formalized in the following proposition (proven in Appendix B.4).

Proposition 8 *Given the assumptions stated in Theorem 6, the generalized curvature $k_f(\mathcal{S})$ for any set $\mathcal{S}$ satisfies* $\kappa_f(\mathcal{S}) \leq \widehat{\kappa}_f = 1 - \frac{\ell^*}{(CQ+1)y_{\max}^2}$.

4. The SELCON algorithm

In this section, we design SELCON, an iterative approximation algorithm to minimize $f(\mathcal{S})$, by leveraging the semi-differential based approach proposed by Iyer et al. (2013a). However, they only consider submodular optimization problems having access to an exact measurement of the objective. In contrast, SELCON works for α-submodular functions and enjoys an approximation guarantee even when it can only access an imperfect estimate of the learned parameters.

4.1. Outline of SELCON

At the very outset, SELCON is an iterative Majorization-Minimization algorithm for minimizing a monotone α-submodular function. We first develop a modular upper bound of $f(\mathcal{S})$. Then, at each iteration, we minimize this upper bound and refine the estimate of the candidate set $\mathcal{S}$.

Modular upper bound of $f(\mathcal{S})$. Given an α-submodular function f and a fixed set $\widehat{\mathcal{S}}$, we can obtain the modular upper bound of $f(\mathcal{S})$, as follows (see details in Appendix C.1).

Lemma 9 *Given a fixed set $\widehat{\mathcal{S}}$ and an α-submodular function $f(\mathcal{S})$, let the modular function $m_{\widehat{\mathcal{S}}}^f[\mathcal{S}]$ be defined as follows:*

$$m_{\widehat{\mathcal{S}}}^f[\mathcal{S}] = f(\widehat{\mathcal{S}}) - \sum_{i\in\widehat{\mathcal{S}}} \alpha f(i|\widehat{\mathcal{S}}\backslash\{i\}) + \sum_{i\in\widehat{\mathcal{S}}\cap\mathcal{S}} \alpha f(i|\widehat{\mathcal{S}}\backslash\{i\}) + \sum_{i\in\mathcal{S}\backslash\widehat{\mathcal{S}}} \frac{f(i|\emptyset)}{\alpha}. \quad (14)$$

Then, $f(\mathcal{S}) \leq m_{\widehat{\mathcal{S}}}^f[\mathcal{S}]$ for all $\mathcal{S} \subseteq \mathcal{D}$.

Note that when $\alpha = 1$, *i.e.*, f is submodular, the expression $m_{\widehat{\mathcal{S}}}^f[\mathcal{S}]$ coincides with the existing modular upper bounds for submodular functions (Nemhauser et al., 1978; Iyer et al., 2013a; Iyer & Bilmes, 2012). Given a $\widehat{\mathcal{S}}$, $m_{\widehat{\mathcal{S}}}^f[\mathcal{S}]$ is modular in $\mathcal{S}$. Therefore, as suggested by Eq. (14), in order to minimize this modular upper bound m with respect to a k-member set $\mathcal{S}$, we need to compute the last two terms, *i.e.*, $\widehat{\alpha}_f f(i|\widehat{\mathcal{S}}\backslash\{i\})$ for all $i \in \widehat{\mathcal{S}}$ and, $f(i|\emptyset)/\widehat{\alpha}_f$ for all $i \notin \widehat{\mathcal{S}}$; and finally, choose the k smallest elements based on these quantities.

Algorithm 1 SELCON Algorithm

Require: Training data $\mathcal{D}$, λ, $\widehat{\alpha}_f$, initial subset $\mathcal{S}_0$ of size k initial model parameters.

1: $\widehat{\mathcal{S}} \leftarrow \mathcal{S}_0$
2: **for all** $i \in \mathcal{D}$ **do**
3: $\quad(\widehat{\boldsymbol{w}}, \widehat{\boldsymbol{\mu}}), \widehat{f}(\{i\}) \leftarrow \text{Train}(F(\boldsymbol{w}, \boldsymbol{\mu}, \{i\}))$
4: **end for**
5: **for** $l \in [L]$ **do**
6: $\quad(\widehat{\boldsymbol{w}}, \widehat{\boldsymbol{\mu}}), \widehat{f}(\widehat{\mathcal{S}}) \leftarrow \text{Train}(F(\boldsymbol{w}, \boldsymbol{\mu}, \widehat{\mathcal{S}}))$
7: $\quad$**for all** $i \in \widehat{\mathcal{S}}$ **do**
8: $\quad\quad\widehat{f}(\widehat{\mathcal{S}} \setminus \{i\}) \leftarrow \text{Train}(F(\boldsymbol{w}, \boldsymbol{\mu}, \widehat{\mathcal{S}}\backslash\{i\}))$
9: $\quad\quad m[i] \leftarrow \widehat{\alpha}_f[\widehat{f}(\widehat{\mathcal{S}}) - \widehat{f}(\widehat{\mathcal{S}} \setminus \{i\})]$
10: $\quad$**end for**
11: $\quad$For all $i \notin \widehat{\mathcal{S}}$, set $m[i] = \widehat{f}(i \,|\, \emptyset)/\widehat{\alpha}_f$
12: $\quad$Pick the k smallest elements from $\{m[i]|\}_{i\in\mathcal{D}}$ to update $\widehat{\mathcal{S}}$
13: $\quad\mathcal{S}^{(l)} \leftarrow \widehat{\mathcal{S}}$
14: **end for**
15: Return $\widehat{\mathcal{S}}, \widehat{\boldsymbol{w}}, \widehat{\boldsymbol{\mu}}$

The iterative procedure. We summarize SELCON in Algorithm 1. Given the current estimate of the candidate set $\widehat{\mathcal{S}}$, SELCON computes $\widehat{\alpha}_f \widehat{f}(i|\widehat{\mathcal{S}}\backslash\{i\})$ for $i \in \widehat{\mathcal{S}}$ in line 9 and $\widehat{f}(i|\emptyset)/\widehat{\alpha}_f$ for $i \notin \widehat{\mathcal{S}}$ in line 11. The algorithm next picks the k smallest values in line 12 to minimize m and update $\widehat{\mathcal{S}}$. Note that computation of f here requires an estimate of the model parameters $\boldsymbol{w}$ and the Lagrangian multipliers $\boldsymbol{\mu}$. However, a training algorithm might only provide a noisy or imperfect estimate of these parameters. Hence, we can only compute $\widehat{f}(\bullet)$, an imperfect estimate of $f(\bullet)$. Appendix C.4 presents the convergence properties of SELCON.

4.2. Approximation guarantee

We now show that SELCON admits a bounded approximation guarantee in the case of both perfect and imperfect estimates of the parameters $(\widehat{\boldsymbol{w}}, \widehat{\boldsymbol{\mu}})$.

Results with perfect parameter estimates. In the following, we present our first result on the approximation guarantee (proof in Appendix C.2).

Theorem 10 *If the training algorithm in Algorithm 1 (lines 3, 6, 8) provides perfect estimates of the model parameters, it obtains a set $\widehat{\mathcal{S}}$ which satisfies:*

$$f(\widehat{\mathcal{S}}) \leq \frac{k}{\widehat{\alpha}_f(1+(k-1)(1-\widehat{\kappa}_f)\widehat{\alpha}_f)} f(\mathcal{S}^*) \quad (15)$$

where $\widehat{\alpha}_f$ and $\widehat{\kappa}_f$ are as stated in Theorem 6 and Proposition 8 respectively.

Results with imperfect parameter estimates. Data-driven training algorithms may not provide the optimal value of model parameters, even if the underlying loss function is convex. Therefore, in practice, SELCON can only access an imperfect estimate of $\widehat{\boldsymbol{w}}, \widehat{\boldsymbol{\mu}}$ in lines 3, 6 and 8. Submodular and weakly submodular optimization in the presence of imperfect estimates has been widely studied in literature (Qian et al., 2017; El Halabi & Jegelka, 2020; Hassidim & Singer, 2018; 2017; Horel & Singer, 2016; Singla et al., 2016). However, to the best of our knowledge, they do not tackle the problem of cardinality-constrained minimization of an α-submodular function. In this context, a notable contribution of our work is that, SELCON also enjoys a relaxed approximation guarantee in these cases, which renders it practically useful. We formally state the result as follows (proven in Appendix C.3).

Theorem 11 *If the training algorithm (lines 3, 6, 8) in Algorithm 1 provides imperfect estimates, so that* $\|F(\widehat{\boldsymbol{w}}, \widehat{\boldsymbol{\mu}}, \mathcal{S}) - F(\boldsymbol{w}^*(\boldsymbol{\mu}^*(\mathcal{S}), \mathcal{S}), \boldsymbol{\mu}^*(\mathcal{S}), \mathcal{S})\| \leq \epsilon$ *for any* $\mathcal{S}$*, then Algorithm 1 obtains a set* $\widehat{\mathcal{S}}$ *that satisfies:*

$$f(\widehat{\mathcal{S}}) \leq \left(\frac{k}{\widehat{\alpha}_f(1 + (k-1)(1-\widehat{\kappa}_f)\widehat{\alpha}_f)} + \frac{2k\epsilon}{\ell} \right) f(S^*),$$

where $\ell = \min_{i \in \mathcal{D}} \min_{\boldsymbol{w}} \lambda ||\boldsymbol{w}||^2 + (y_i - h_{\boldsymbol{w}}(\boldsymbol{x}_i))^2$*,* $\widehat{\alpha}_f$ *and* $\widehat{\kappa}_f$ *are obtained in Theorem 6 and Proposition 8, respectively.*

Discussion on the approximation ratio. A trite calculation shows that, for the regime of λ defined in Theorem 6 and a small value of ϵ, the approximation ratio of SELCON is $O(y_{\max}^4/y_{\min}^4)$. While such a ratio may appear to be conservative, there are several applications such as house price prediction or stock prediction, where $y_{\max}/y_{\min}$ may not be too high. Apart from that, one can always pre-process the dataset by adding an offset to y and augmenting a constant in the feature $\boldsymbol{x}$, to control this ratio, as illustrated in Appendix E. Moreover, since our approximation ratio holds for any monotone α-submodular function with bounded curvature, it can be of independent technical interest.

5. Experiments

In this section, we present experimental results and analysis on several real-world datasets to evaluate the performance of SELCON against several competitive baselines. Thereafter, we show that our framework is also practically useful in a fair regression setup, where the validation loss bounds are used to ensure that the error for each protected group is below an acceptable level of threshold. Appendix E contains additional experiments.

5.1. Experimental setup

Datasets. We experiment with five real world datasets, *viz.*, Cadata (16718 instances), Law (20800 instances), NYSE-High (701348 instances), NYSE-Close (701348 instances), and Community-and-crime (1994 instances), all briefly described in Appendix D.

Baselines. We compare SELCON against seven baselines. (1) *Full-selection:* It uses full data for training without any validation error constraint. (2) *Full-with-constraints:* It uses full data for training, subject to the same validation error constraints used in SELCON. (3) *Random-selection:* It samples a training subset uniformly at random, but it does not employ any constraint on validation set. (4) *Random-with-constraints:* It is the same as Random-selection, except that it uses the constraints on validation set errors. (5) *CRAIG (Mirzasoleiman et al., 2020):* This is a coreset based data selection method, that however, does not use any constraint on the validation set. (6) *GLISTER (Killamsetty et al., 2021b):* This is a data selection method that uses validation set to fine tune the trained model, which however, does not pose any explicit constraint on the validation set error. (7) SELCON-*without-constraint:* Here, we solve the optimization problem (2), without the validation error constraints.

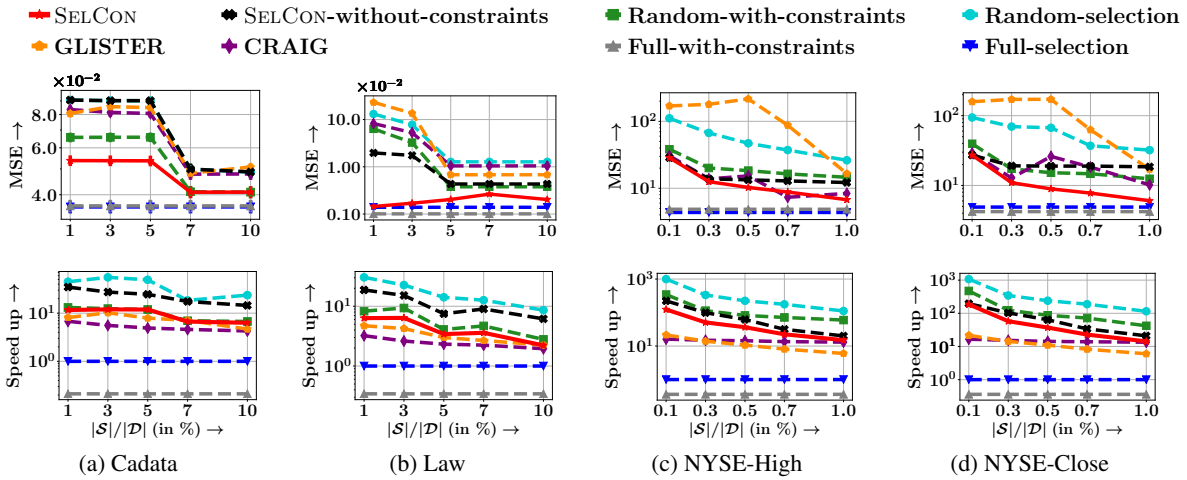
Implementation details. In Algorithm 1, if we set the number of epochs for Train( ) in line 6 to T, this training routine runs for $N = LT$ epochs, where L is the number of iterations of the for-loop (lines 5–14). To make a fair comparison, we used the same number of epochs N and the same batch size b across all baselines and SELCON for training the underlying model. Specifically, we set $N = 2000$ for Cadata and Law , $N = 5000$ for the NYSE datasets; and, $b = \min\{|\mathcal{S}|, 1000\}$ across all datasets. Additionally, SELCON involves two more sets of small scale optimization problems (lines 3 and 8 respectively), where we set the number of epochs as 3. Moreover for the optimization of $f(\mathcal{S}\backslash\{i\})$ in line 8, we use the same batch size $b = \min\{|\mathcal{S}|, 1000\}$ as stated earlier. In each experiment, we used (random) 89% training, 1% validation and 10% test folds. We employed pytorch with the adam optimizer for all experiments. Further details about the implementation are provided in Appendix D.

5.2. Predictive performance and efficiency

We evaluate the performance of each data selection method in terms of the mean squared error (MSE) $\mathbb{E}[(y - \hat{y})^2]$ on the test set. We also compute the computational efficiency of a method in terms of the speed-up it achieves with respect to Full-selection, *i.e.*, $\text{RunTime}_{\text{Full-selection}}/\text{RunTime}_{\text{method}}$, where $\text{RunTime}_\bullet$ is time taken by the corresponding method to complete both the subset selection and model training. Here, we constrain the total loss on the validation set, *i.e.*, we set $Q = 1$.

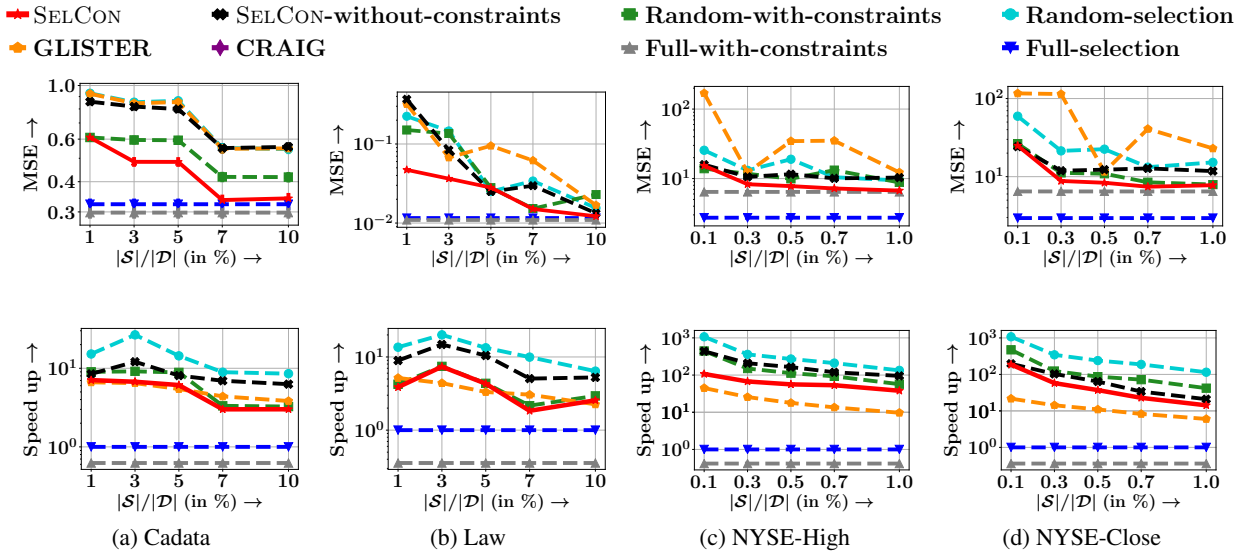
Linear regression. Here, we compare the performance of SELCON for linear regression ($h_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$) against all the baselines across the first four datasets[4], described in

[4] Due to its small size, we ignore Community-and-crime in this experiment.



(a) Cadata (b) Law (c) NYSE-High (d) NYSE-Close

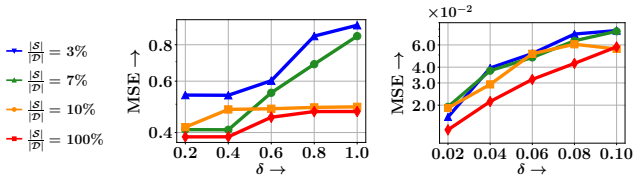
Figure 1. Variation of performance in terms of the mean squared error (MSE, top row) and the computational efficiency in terms of speed up with respect to Full-selection (bottom row) for all methods, *i.e.*, SELCON (Algorithm 1), SELCON-without-constraints, Random-selection, Random-with-constraints, Full-selection, Full-with-constraints, CRAIG (Mirzasoleiman et al., 2020) and GLISTER (Killamsetty et al., 2021b) across different datasets with 10% held-out set and 1% validation set. We set the number of partitions $Q = 1$.



(a) Cadata (b) Law (c) NYSE-High (d) NYSE-Close

Figure 2. Variation of performance for nonlinear regression using $\boldsymbol{h}_w(\boldsymbol{x}) = \boldsymbol{w}^\top \text{ReLU}(\boldsymbol{W}\boldsymbol{x})$, in terms of the mean squared error (MSE, top row) and the computational efficiency in terms of speed up with respect to Full-selection (bottom row) for all methods, *i.e.*, SELCON (Algorithm 1), SELCON-without-constraints, Random-selection, Random-with-constraints, Full-selection, Full-with-constraints, CRAIG (Mirzasoleiman et al., 2020) and GLISTER (Killamsetty et al., 2021b) across different datasets with 10% held-out set and 1% validation set. We set the number of partitions $Q = 1$.

Section 5.1. Moreover, for smaller datasets, *i.e.*, Cadata and Law, we consider $|\mathcal{S}|/|\mathcal{D}| \in [0.01, 0.1]$, whereas, for larger datasets, *i.e.*, NYSE-High and NYSE-Close, we consider $|\mathcal{S}|/|\mathcal{D}| \in [0.001, 0.01]$. In Figure 1, we summarize the results. We make the following observations. (i) SELCON shows better predictive accuracy than all the baselines except Full-selection and Full-with-constraints in most of the cases. The performance of Random-with-constraints is often comparable with SELCON especially when $|\mathcal{S}|/|\mathcal{D}|$ is too high ($> 5\%$ in Cadata and Law) or too low ($< 0.3\%$ in NYSE datasets). On the Law dataset, SELCON's performance is noteworthy - with 1% training data, it performs at par with Full-selection. In most cases, the performance gain provided by SELCON over Random-with-constraints is statistically significant (Wilcoxon signed-rank test, p-value $= 0.05$) while SELCON consistently outperforms the other baselines. (ii) SELCON shows a significant speed up with respect to Full-with-constraints, Full-selection, GLISTER and CRAIG. In fact, with 1% subset size, SELCON shows a $10\times$ speed up with respect to Full-selection, often with neg-



(a) Cadata (b) Law

Figure 3. Variation of mean squared error (MSE) across different values of validation error bound δ for different sizes of $|\mathcal{S}|$. We observe that for different values of $|\mathcal{S}|$, the performance generally improves as δ decreases.

ligible loss in accuracy (see Law and NYSE-Close). However, SELCON is slower than Random-selection, Random-with-constraints and SELCON-without-constraint. This is because SELCON-without-constraint does not have any validation loss constraints; and, none of the random heuristics involves any additional overhead time due to subset selection. (iii) CRAIG and GLISTER do not involve any explicit validation set constraints, which often curbs their predictive power. On the other hand, even Random-with-constraints is able to outperform them in terms of the predictive performance, which is because of their improved generalization ability due to the presence of the explicit validation error constraints.

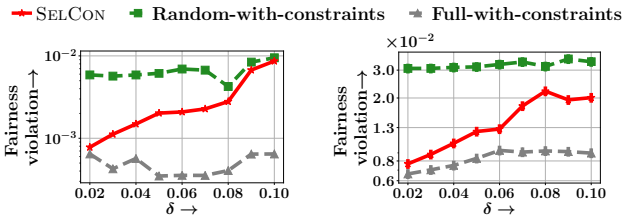
Nonlinear regression. Next, we analyze the performance and efficiency of SELCON, when $\boldsymbol{h}_w(\boldsymbol{x}) = \boldsymbol{w}^\top \text{ReLU}(\boldsymbol{W}\boldsymbol{x})$. In Figure 2, we summarize the results[5] which shows that SELCON can trade off between efficiency and performance more effectively than the baselines (results similar to linear regression).

Effect of δ. We next investigate the effect of δ on the predictive performance for different sizes of $|\mathcal{S}|$. In Figure 3, we summarize the results for linear models, which shows that for different values of $|\mathcal{S}|$, the performance generally improves as δ decreases.

5.3. Application to fair regression

Fairness in regression requires that the prediction error limited to any protected group is below a pre-specified label (Agarwal et al., 2019) and therefore, such an application naturally fits in our setting. To that end, we apply our approach to the Law and Community-and-crime datasets (Agarwal et al., 2019) and enforce fairness with respect to the race of an individual as the protected attribute. More specifically, given Q types of races $\{r_q\}_{q\in[Q]}$ ($Q = 8$ in Law, $Q = 4$ in Community-and-crime) in the dataset, we partition the validation set $\mathcal{V}$ into the subsets $V_1, .., V_Q$, so that each subset V_q consists of individuals with the race r_q, *i.e.*, $V_q = \{(\boldsymbol{x}_j, y_j) \,|\, \text{Race of individual } j = r_q\}$.

Results. In Figure 4, we plot the performance of SELCON in terms of the mean fairness violation, as measured by $\mathbb{E}[|(y_i - \hat{y}_i)^2 - (y_j - \hat{y}_j)^2| \,|\, i \in V_q, j \in \mathcal{V}\backslash V_q]$ for various values of δ. We compare SELCON's performance against Full-with-constraints and Random-with-constraints, the only other methods that can enforce fairness by means of error constraints on the validation set. Evidently, SELCON guarantees fairness more effectively than Random-with-constraints. Moreover, for low values of δ, the performance of SELCON is close to Full-with-constraints.



(a) Law (b) Community-and-crime

Figure 4. Data selection in fair regression - These plots show the variation of fairness violation measured in terms of $\mathbb{E}[|(y_i - \hat{y}_i)^2 - (y_j - \hat{y}_j)^2| \,|\, i \in V_q, j \in \mathcal{V}\backslash V_q]$ with δ for $|\mathcal{S}| = 0.1|\mathcal{V}|$. Here, V_q consists of individuals with a particular race r_q. We observe that SELCON guarantees fairness more effectively than Random-with-constraints.

6. Conclusion

We presented a novel data subset selection formulation that aims to select a subset $\mathcal{S}$ by controlling the generalization errors. Specifically, we considered L_2 regularized regression over candidate training set $\mathcal{S}$, subject to the error bounds on different partitions of the validation set. Such error bounds reduce the generalization error that could otherwise increase owing to training on a small sized data. Thereafter, we reformulated our data selection task as a new optimization problem and showed its equivalence to minimization of a monotone and α-submodular function. Finally, we designed a majorization-minimization based approximation algorithm SELCON to solve this problem in the face of imperfect training. Our experiments show that SELCON can more effectively trade off between accuracy and efficiency than several baselines. Our work opens several areas for future work; *e.g.*, it can be extended to data selection for classification as well as data removal for outlier detection.

Acknowledgements

We thank anonymous reviewers for providing constructive feedback. Durga Sivasubramanian is supported by a the Prime Minisiter Research Fellowship. Ganesh and Abir are also grateful to IBM Research, India (specifically the IBM AI Horizon Networks - IIT Bombay initiative) for their support and sponsorship. Abir also acknowledges the DST Inspire Award and IITB Seed Grant. Rishabh acknowledges support from the UT Dallas startup grant.

[5] We omitted the results for CRAIG in nonlinear regression because the data selection component of CRAIG needs to be run for several epochs for non-convex losses (Mirzasoleiman et al., 2020), and hence, it did not scale for the large datasets.

References

Agarwal, A., Dudik, M., and Wu, Z. S. Fair regression: Quantitative definitions and reduction-based algorithms. In *International Conference on Machine Learning*, pp. 120–129. PMLR, 2019.

Bairi, R., Iyer, R., Ramakrishnan, G., and Bilmes, J. Summarization of multi-document topic hierarchies using submodular mixtures. In *ACL*, pp. 553–563, 2015.

Bhatia, K., Jain, P., Kamalaruban, P., and Kar, P. Consistent robust regression. In *Advances in Neural Information Processing Systems*, pp. 2110–2119, 2017.

Bian, A. A., Buhmann, J. M., Krause, A., and Tschiatschek, S. Guarantees for greedy maximization of non-submodular functions with applications. *arXiv preprint arXiv:1703.02100*, 2017.

Boutsidis, C., Drineas, P., and Magdon-Ismail, M. Near-optimal coresets for least-squares regression. *IEEE transactions on information theory*, 59(10):6880–6892, 2013.

Campbell, T. and Broderick, T. Bayesian coreset construction via greedy iterative geodesic ascent. In *International Conference on Machine Learning*, pp. 698–706, 2018.

Casella, G. and Berger, R. L. *Statistical inference*, volume 2. Duxbury Pacific Grove, CA, 2002.

Clarkson, K. L. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. 2010.

Das, A. and Kempe, D. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975*, 2011.

De, A., Koley, P., Ganguly, N., and Gomez-Rodriguez, M. Regression under human assistance. *AAAI*, 2020.

De, A., Okati, N., Zarezade, A., and Gomez-Rodriguez, M. Classification under human assistance. *AAAI*, 2021.

El Halabi, M. and Jegelka, S. Optimal approximation for unconstrained non-submodular minimization. In *International Conference on Machine Learning*, pp. 3961–3972. PMLR, 2020.

Gatmiry, K. and Gomez-Rodriguez, M. Non-submodular function maximization subject to a matroid constraint, with applications. *arXiv preprint arXiv:1811.07863*, 2018.

Har-Peled, S. and Mazumdar, S. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pp. 291–300, 2004.

Hashemi, A., Ghasemi, M., Vikalo, H., and Topcu, U. Submodular observation selection and information gathering for quadratic models. *arXiv preprint arXiv:1905.09919*, 2019.

Hassani, H., Soltanolkotabi, M., and Karbasi, A. Gradient methods for submodular maximization. In *Advances in Neural Information Processing Systems*, pp. 5841–5851, 2017.

Hassidim, A. and Singer, Y. Submodular optimization under noise. 2017.

Hassidim, A. and Singer, Y. Optimization for approximate submodularity. pp. 394–405. Curran Associates Inc., 2018.

Horel, T. and Singer, Y. Maximization of approximately submodular functions. 2016.

Iyer, R. and Bilmes, J. Algorithms for approximate minimization of the difference between submodular functions, with applications. *arXiv preprint arXiv:1207.0560*, 2012.

Iyer, R. and Bilmes, J. Polyhedral aspects of submodularity, convexity and concavity. *arXiv preprint arXiv:1506.07329*, 2015.

Iyer, R., Jegelka, S., and Bilmes, J. Fast semidifferential-based submodular function optimization: Extended version. In *ICML*, 2013a.

Iyer, R. K., Jegelka, S., and Bilmes, J. A. Curvature and optimal algorithms for learning and minimizing submodular functions. In *NeurIPS*, 2013b.

Kaushal, V., Iyer, R., Kothawade, S., Mahadev, R., Doctor, K., and Ramakrishnan, G. Learning from less data: A unified data subset selection and active learning framework for computer vision. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1289–1299. IEEE, 2019.

Killamsetty, K., Sivasubramanian, D., Mirzasoleiman, B., Ramakrishnan, G., De, A., and Iyer, R. Grad-match: A gradient matching based data subset selection for efficient learning. *arXiv preprint arXiv:2103.00123*, 2021a.

Killamsetty, K., Subramanian, D., Ramakrishnan, G., and Iyer, R. Glister: A generalization based data selection framework for efficient and robust learning. *In AAAI*, 2021b.

Kirchhoff, K. and Bilmes, J. Submodularity for data selection in machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 131–141, 2014.

Kuhnle, A., Smith, J. D., Crawford, V. G., and Thai, M. T. Fast maximization of non-submodular, monotonic functions on the integer lattice. *arXiv preprint arXiv:1805.06990*, 2018.

Kulkarni, A., Uppalapati, N. R., Singh, P., and Ramakrishnan, G. An interactive multi-label consensus labeling model for multiple labeler judgments. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI), 2018*, pp. 1479–1486. AAAI Press, 2018.

Lehmann, B., Lehmann, D., and Nisan, N. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

Liu, Y., Iyer, R., Kirchhoff, K., and Bilmes, J. Svitchboard ii and fisver i: High-quality limited-complexity corpora of conversational english speech. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

Lucic, M., Faulkner, M., Krause, A., and Feldman, D. Training gaussian mixture models at scale via coresets. *The Journal of Machine Learning Research*, 18(1):5885–5909, 2017.

Mirzasoleiman, B., Bilmes, J., and Leskovec, J. Coresets for data-efficient training of machine learning models. *In Proc. ICML*, 2020.

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.

Pace, R. K. and Barry, R. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997.

Qian, C., Shi, J.-C., Yu, Y., Tang, K., and Zhou, Z.-H. Subset selection under noise. pp. 3560–3570, 2017.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343, 2018.

Rothenhäusler, D., Meinshausen, N., Bühlmann, P., and Peters, J. Anchor regression: heterogeneous data meets causality. *arXiv preprint arXiv:1801.06229*, 2018.

Sener, O. and Savarese, S. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Singla, A., Tschiatschek, S., and Krause, A. Noisy submodular maximization via adaptive sampling with applications to crowdsourced image collection summarization. 2016.

Wei, K., Iyer, R., and Bilmes, J. Fast multi-stage submodular maximization. In *International conference on machine learning*, pp. 1494–1502. PMLR, 2014a.

Wei, K., Liu, Y., Kirchhoff, K., and Bilmes, J. Unsupervised submodular subset selection for speech data. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4107–4111. IEEE, 2014b.

Wei, K., Iyer, R., and Bilmes, J. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pp. 1954–1963, 2015.

Wightman. Lsac national longitudinal bar passage study. 1998.

Wilcoxon, F. Individual comparisons by ranking methods. In *Breakthroughs in statistics*, pp. 196–202. Springer, 1992.

Zhang, H. and Vorobeychik, Y. Submodular optimization with routing constraints. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pp. 8778–8788, 2018.