Adapting SELCON for Efficient Data Subset Selection to assist Classification Tasks

Course Project - CS769 *Optimization for Machine Learning*

MOHAMAD HASSAN N C [1], NINAD GANDHI [1], PRABHAT REDDY [1], NITISH KUMAR [2]

[1]MS by Research, Centre for Machine Intelligence and Data Science, IITB
[2]M.Tech, Electrical Engineering, IITB

May 4, 2023

Table of contents

Introduction

Problem Statement

- Computational complexity involved in processing and analysing.

Problem Statement

- Computational complexity involved in processing and analysing.
- Requires more computational time and power consumption

Problem Statement

- Computational complexity involved in processing and analysing.
- Requires more computational time and power consumption
- Traditional training methods are often less power-efficient, requiring large amounts of memory and storage to process large datasets.

Data Subset Selection

- Make training more efficient by selecting a **subset** of the training data that performs as well as the full dataset.

Data Subset Selection

- Make training more efficient by selecting a **subset** of the training data that performs as well as the full dataset.
- Choose some points from the original dataset and the objectives might be to capture **diversity or importance**.

Data Subset Selection

- Make training more efficient by selecting a **subset** of the training data that performs as well as the full dataset.
- Choose some points from the original dataset and the objectives might be to capture **diversity or importance**.
- **Coreset selection** aims to find a good representative set that captures important structure and information about the original dataset.

Data Subset Selection

- Make training more efficient by selecting a **subset** of the training data that performs as well as the full dataset.
- Choose some points from the original dataset and the objectives might be to capture **diversity or importance**.
- **Coreset selection** aims to find a good representative set that captures important structure and information about the original dataset.
- Uses more sophisticated algorithms such as **k-means clustering or sampling**.

Why is it hard?

- **Balancing tradeoffs**
- **Incorporating diversity**
- **Incorporating fairness**
- **Dealing with large datasets**

Approach

- **Fairness** is imposed in the choice of points selected in the data subset for each of the protected groups in the **validation sets**.

Approach

- **Fairness** is imposed in the choice of points selected in the data subset for each of the protected groups in the **validation sets**.
- We will discuss fair regression with bounded group loss and provides a reduction to the classification problem.

Approach

- **Fairness** is imposed in the choice of points selected in the data subset for each of the protected groups in the **validation sets**.
- We will discuss fair regression with bounded group loss and provides a reduction to the classification problem.
- Objective is to **limit the error** for a validation group.

Approach

- **Fairness** is imposed in the choice of points selected in the data subset for each of the protected groups in the **validation sets**.
- We will discuss fair regression with bounded group loss and provides a reduction to the classification problem.
- Objective is to **limit the error** for a validation group.
- **We adapt SELCON, a data subset selection algorithm for L2 regularized regression tasks, to the binary classification setting.**

Problem Formulation

Notation

- $\{\boldsymbol{x}_i, y_i\}_{i \in D}$ denotes the set of training samples
- $\{\boldsymbol{x}_j, y_j\}_{j \in V}$ denote the set of validation samples
  - where, $\boldsymbol{x} \in \mathbb{R}^d$ and $y \in \{0, 1\}$

Notation

- $\{\boldsymbol{x}_i, y_i\}_{i \in D}$ denotes the set of training samples
- $\{\boldsymbol{x}_j, y_j\}_{j \in V}$ denote the set of validation samples
  - where, $\boldsymbol{x} \in \mathbb{R}^d$ and $y \in \{0, 1\}$
- The validation set is split into Q partition subsets
  - Where, $V = V_1 \cup V_2 \cup ... \cup V_Q$

Notation

- $\{\boldsymbol{x}_i, y_i\}_{i \in D}$ denotes the set of training samples
- $\{\boldsymbol{x}_j, y_j\}_{j \in V}$ denote the set of validation samples
  - where, $\boldsymbol{x} \in \mathbb{R}^d$ and $y \in \{0, 1\}$
- The validation set is split into Q partition subsets
  - Where, $V = V_1 \cup V_2 \cup ... \cup V_Q$
- Define a logistic regression model $h_{\boldsymbol{w}} : \mathbb{R}^d \rightarrow [0, 1]$ parameterized by the weights $\boldsymbol{w} \in \mathbb{R}^d$ as follows.

$$h_{\boldsymbol{w}}(\boldsymbol{x}) = \sigma(\boldsymbol{w}^T \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{w}^T \boldsymbol{x}}} \tag{1}$$

Objective Function

We define our regularized constrained objective function,

$$\min_{S \subset \mathcal{D}, \boldsymbol{w}} \quad \sum_{i \in S} [\lambda ||\boldsymbol{w}||^2 + \mathcal{L}(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)]$$

$$\text{subject to,} \quad \frac{\sum_{i \in V_q} \mathcal{L}(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)}{|V_q|} \leq \delta, \forall q \in [Q],$$

$$|S| = k$$

where,

- $\mathcal{L}(\hat{y}, y) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$ is the *Binary Crossentropy (BCE) loss*.

Objective Function

We define our regularized constrained objective function,

$$\min_{S \subset \mathcal{D}, \boldsymbol{w}} \quad \sum_{i \in S} [\lambda ||\boldsymbol{w}||^2 + \mathcal{L}(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)]$$

$$\text{subject to,} \quad \frac{\sum_{i \in V_q} \mathcal{L}(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)}{|V_q|} \leq \delta, \forall q \in [Q],$$

$$|S| = k$$

where,

- $\mathcal{L}(\hat{y}, y) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$ is the *Binary Crossentropy (BCE) loss*.
- λ is the regularizer coefficient.

Objective Function

We define our regularized constrained objective function,

$$\min_{S \subset \mathcal{D}, \boldsymbol{w}} \quad \sum_{i \in S} [\lambda ||\boldsymbol{w}||^2 + \mathcal{L}(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)]$$

$$\text{subject to,} \quad \frac{\sum_{i \in V_q} \mathcal{L}(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)}{|V_q|} \leq \delta, \forall q \in [Q],$$

$$|S| = k$$

where,

- $\mathcal{L}(\hat{y}, y) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$ is the *Binary Crossentropy (BCE) loss*.
- λ is the regularizer coefficient.
- δ is the tolerance of per-sample validation error.

Relaxed Objective

We introduce some slack variables ξ_q, for each subset q, with a constant multiplier C

$$\min_{S \subset \mathcal{D}, \mathbf{w}, \{\xi_q\}_{q \in [Q]}} \quad \sum_{i \in S} [\lambda ||\boldsymbol{w}||^2 + \mathcal{L}(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)] + C \sum_{q \in [Q]} \xi_q$$

$$\text{subject to,} \quad \frac{\sum_{i \in V_q} \mathcal{L}(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)}{|V_q|} \leq \delta + \xi_q, \forall q \in [Q],$$

$$\xi_q \geq 0, \forall q \in [Q] \quad \text{and } |S| = k$$

- C controls the amount of slackness in the objective.

Relaxed Objective

We introduce some slack variables ξ_q, for each subset q, with a constant multiplier C

$$\min_{S \subset \mathcal{D}, \mathbf{w}, \{\xi_q\}_{q \in [Q]}} \quad \sum_{i \in S} [\lambda ||\boldsymbol{w}||^2 + \mathcal{L}(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)] + C \sum_{q \in [Q]} \xi_q$$

$$\text{subject to,} \quad \frac{\sum_{i \in V_q} \mathcal{L}(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)}{|V_q|} \leq \delta + \xi_q, \forall q \in [Q],$$

$$\xi_q \geq 0, \forall q \in [Q] \quad \text{and } |S| = k$$

- C controls the amount of slackness in the objective.
- If $C \to \infty$, then the slack variables must go to 0.

Relaxed Objective

We introduce some slack variables ξ_q, for each subset q, with a constant multiplier C

$$\min_{S \subset \mathcal{D}, \mathbf{w}, \{\xi_q\}_{q \in [Q]}} \quad \sum_{i \in S} [\lambda ||\boldsymbol{w}||^2 + \mathcal{L}(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)] + C \sum_{q \in [Q]} \xi_q$$

$$\text{subject to,} \quad \frac{\sum_{i \in V_q} \mathcal{L}(h_{\boldsymbol{w}}(\boldsymbol{x}_i), y_i)}{|V_q|} \leq \delta + \xi_q, \forall q \in [Q],$$

$$\xi_q \geq 0, \forall q \in [Q] \quad \text{and } |S| = k$$

- C controls the amount of slackness in the objective.
- If $C \to \infty$, then the slack variables must go to 0.
- $C \to \infty$ corresponds to the original constrained objective.

Converting to the Dual Objective

We convert the original problem to a dual version by introducing Lagrange multipliers μ_i and writing the dual objective as

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

Then the optimization problem can be reformulated in terms of F as

$$\max_{\mathbf{0} \leq \boldsymbol{\mu} \leq C\mathbf{1}} \min_{\boldsymbol{w}} F(\boldsymbol{w}, \boldsymbol{\mu}, S)$$

Solving F

We solve the outer maximization problem first to obtain $\boldsymbol{\mu}^*(S)$, then solve the inner minimization problem to obtain $\boldsymbol{w}^*(\boldsymbol{\mu}^*(S), S)$. The solution to dual problem can be written as

$$f(S) = F(\boldsymbol{w}^*(\boldsymbol{\mu}^*(S), S), \boldsymbol{\mu}^*(S), S) \tag{3}$$

This solution $f(S)$ serves as a lower bound to the solution of the primal problem due to weak duality. Therefore, we aim to obtain a subset by solving the following optimization problem.

$$\min_{S \subset D} f(S), \text{ subject to, } |S| = k \tag{4}$$

SELCON for classification

The SELCON algorithm for data subset selection

- The idea behind SELCON is to iteratively minimize an upper bound over $f(S)$ to iteratively improve the estimate of the best subset S.

The SELCON algorithm for data subset selection

- The idea behind SELCON is to iteratively minimize an upper bound over $f(S)$ to iteratively improve the estimate of the best subset S.
- The upper bound $m^f_{\hat{S}}[S]$ is defined as follows.

$$m^f_{\hat{S}}[S] = f(\hat{S}) - \sum_{i \in \hat{S}} \alpha f(i|\hat{S} \setminus \{i\}) + \sum_{i \in \hat{S} \cap S} \alpha f(i|\hat{S} \setminus \{i\}) + \sum_{i \in S \setminus \hat{S}} \frac{f(i|\phi)}{\alpha} \quad (5)$$

The SELCON algorithm for data subset selection

- The idea behind SELCON is to iteratively minimize an upper bound over $f(S)$ to iteratively improve the estimate of the best subset S.
- The upper bound $m^f_{\hat{S}}[S]$ is defined as follows.

$$m^f_{\hat{S}}[S] = f(\hat{S}) - \sum_{i \in \hat{S}} \alpha f(i|\hat{S} \setminus \{i\}) + \sum_{i \in \hat{S} \cap S} \alpha f(i|\hat{S} \setminus \{i\}) + \sum_{i \in S \setminus \hat{S}} \frac{f(i|\phi)}{\alpha} \quad (5)$$

- SELCON has been shown to work for **any set function that is both monotone and α-submodular**.
  - Monotone: $g(a|S) = g(S \cup \{a\}) - g(S) \geq 0$ for $S \subset D$ and $a \in D \setminus S$.
  - α-submodular: $f(a|S) \geq \alpha f(a|T)$ for $S \subseteq T$ and $a \in D \setminus T$.

The SELCON algorithm for data subset selection

- The idea behind SELCON is to iteratively minimize an upper bound over $f(S)$ to iteratively improve the estimate of the best subset S.
- The upper bound $m^f_{\hat{S}}[S]$ is defined as follows.

$$m^f_{\hat{S}}[S] = f(\hat{S}) - \sum_{i \in \hat{S}} \alpha f(i|\hat{S} \setminus \{i\}) + \sum_{i \in \hat{S} \cap S} \alpha f(i|\hat{S} \setminus \{i\}) + \sum_{i \in S \setminus \hat{S}} \frac{f(i|\phi)}{\alpha} \quad (5)$$

- SELCON has been shown to work for **any set function that is both monotone and α-submodular**.
  - Monotone: $g(a|S) = g(S \cup \{a\}) - g(S) \geq 0$ for $S \subset D$ and $a \in D \setminus S$.
  - α-submodular: $f(a|S) \geq \alpha f(a|T)$ for $S \subseteq T$ and $a \in D \setminus T$.
- We show that $f(S)$ in Eq 3 is both monotone and α-submodular.

$f(S)$ is monotone

- Sivasubramanian et al.[1] show that the following holds (proposition 5).

$$f(S \cup a) - f(S) \geq F(\boldsymbol{w}^*(\boldsymbol{\mu}^*(S), S \cup a), \boldsymbol{\mu}^*(S), S \cup a) - F(\boldsymbol{w}^*(\boldsymbol{\mu}^*(S), S \cup a), \boldsymbol{\mu}^*(S), S)$$

- We expand the RHS using Eq. 2 and obtain the following

$$f(S \cup a) - f(S) \geq \lambda \left\| \boldsymbol{w}^*(\boldsymbol{\mu}^*(S), S \cup a) \right\|^2 + \mathcal{L}(h_{\boldsymbol{w}}(x_a), y_a) \geq 0$$

thereby showing $f(S)$ is monotone.

$f(S)$ is α-submodular

Adapted proposition 7 from [1] to the classification setting.

$f(S)$ is α-submodular

Adapted proposition 7 from [1] to the classification setting. Given $0 < y_{min} < |y| < y_{max}$ and h_w is H-lipschitz, we set the regularisation constant as $\lambda \geq \max\{1, 16(1+CQ)^2 y_{max}^2 / I^*\}$. The eigenvalue of the hessian has a finite upper bound say χ_{max}^2, let $l_a(w) = \lambda||w||^2 + L(y_a, \hat{y_a}), \bar{w} = argmin_w l_a(w)$ and $I^* = min_{a \in D} min_w \chi_{max}^2 ||w||^2 + L(y_a, \hat{y_a})$. Then f(S) is α-submodular set function where

$$\alpha \geq \hat{\alpha}_f = 1 - \frac{16(1+CQ)^2 y_{max}^2}{\lambda I^*}$$

$f(S)$ is α-submodular

Adapted proposition 7 from [1] to the classification setting. Given $0 < y_{min} < |y| < y_{max}$ and h_w is H-lipschitz, we set the regularisation constant as $\lambda \geq \max\{1, 16(1+CQ)^2 y_{max}^2 / I^*\}$. The eigenvalue of the hessian has a finite upper bound say χ_{max}^2, let $l_a(w) = \lambda||w||^2 + L(y_a, \hat{y_a}), \bar{w} = argmin_w l_a(w)$ and $I^* = min_{a \in D} min_w \chi_{max}^2 ||w||^2 + L(y_a, \hat{y_a})$. Then f(S) is α-submodular set function where

$$\alpha \geq \hat{\alpha}_f = 1 - \frac{16(1+CQ)^2 y_{max}^2}{\lambda I^*}$$

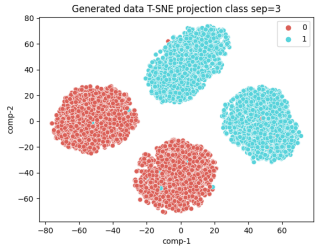
Key step is identifying that the logistic function is lipschitz with constant 1. This removes the need for $x_{\max}$ and λ is modified accordingly.

Experiments

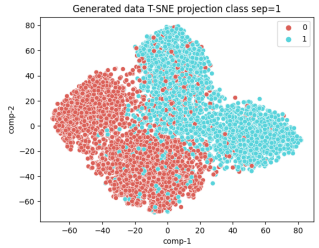
Datasets

- **Madelon dataset**: artificial dataset from NIPS 2003 feature selection challenge, 4400 instances.
- **Gisette dataset**: handwritten digit recognition dataset from NIPS 2003 feature selection challenge, 13500 instances.
- **Toy dataset** generated by using the sklearn's `make_classification` function.
    - We vary the class seperation parameter in the make classification function to generate different datasets.

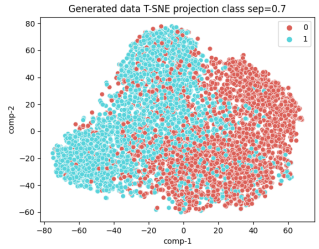
Datasets



(a) sep=3

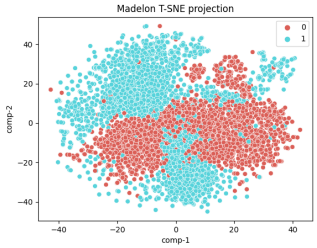


(b) sep=1

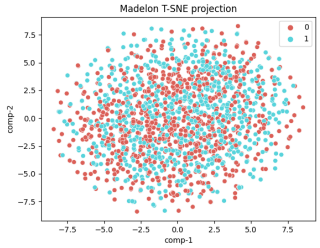


(c) sep=0.7

Figure: T-sne plots for generated dataset for each class seperation



(a) Gisette Dataset



(b) Madelon Dataset

Figure: T-sne plots for NIPS 2003 dataset

Baselines

- **Full selection**: Here we use the entire training data for training the logistic regression and evaluate the performance on the test set.
- **Random subset selection**: Here we pick a training subset uniformly at random and then train the logistic regression model on it

This two baselines are compared against the subset that is returned by the SELCON algorithm.

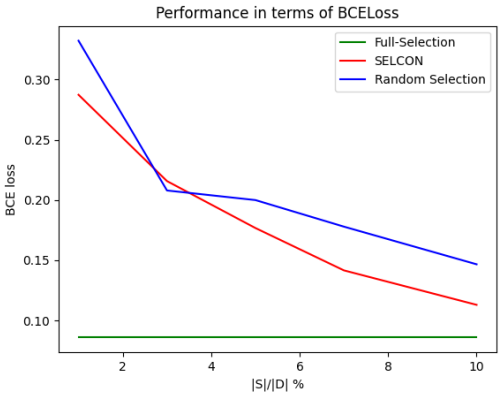
Experimental Setup

- In all of the experiments on the datasets, we have set the number of epochs as 1000
- For Madelon and Gisette Data Set, we have split the official training set into 80% training set and 20% validation set. The official validation set is taken as the test set.
- The number of epochs in the inner optimization is taken as 3.

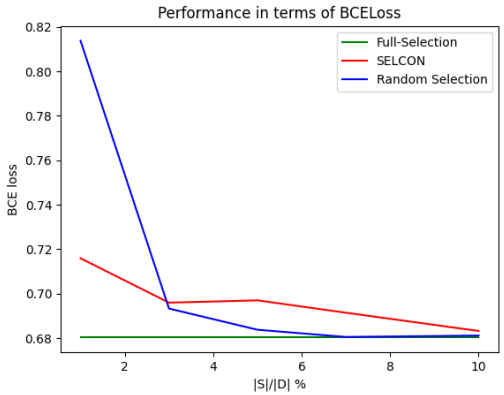
Experimental Setup

- We have used the *Adam optimizer* for the purpose of training the model.
- The Logistic Regression models that are trained for evaluation using the subsets are trained for 100 epochs and the learning rate for the model is set as 0.1.
- The result for the random subset is obtained by averaging the result across 100 different random subsets for the generated data and the Gisette dataset, 50 for the Madelon dataset.

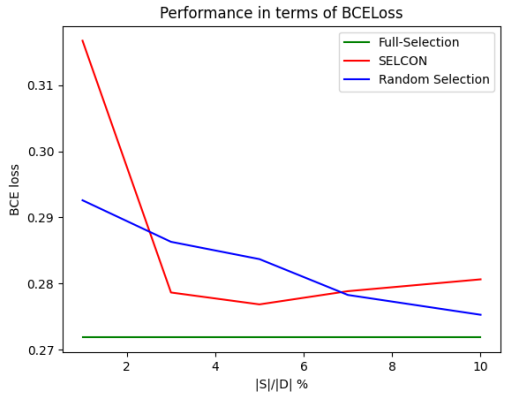
Result



(a) Gisette Dataset

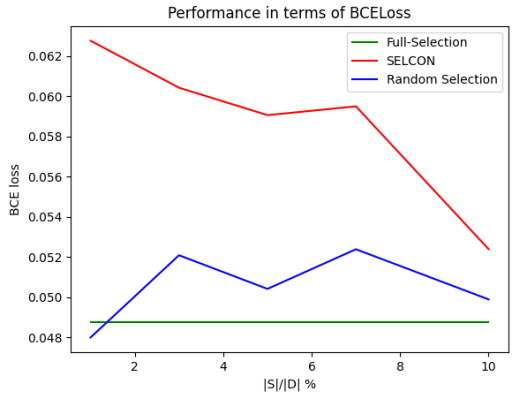


(b) Madelon Dataset

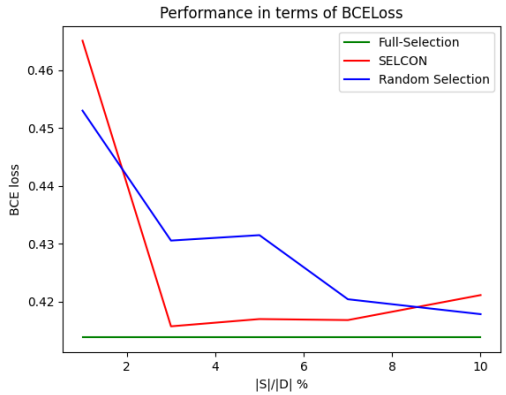


(c) Artificial (class sep=1)

Figure: Gisette, Madelon and Artificial Dataset



(a) Artificial (class sep=3)



(b) Artificial (class sep=0.7)

Discussion on result

- When the classes are well separated and classification task is easy then we find that there is not much benefit of using the SELCON algorithm as random subset selection is achieving similar or lower loss.
- But when the classification task becomes harder and classes are not well separated, SELCON algorithm is found to have the upper hand over the random subset selection.
- We have also seen that when the subset size is around 10% of the total dataset size, the test accuracy is very similar to the model that was trained on the full training set.

Future Directions

- Current work is limited to the task of binary classification. This work can also be extended to multi-classification tasks where BCE loss can be replaced with cross-entropy loss along with slight modifications in the architecture of the model.
- Moreover, there can be another instance in which we can introduce another submodular function such as dispersion min-sum, which can ensure diversity among points that are picked up in the subset.

Thank you!

Questions?

References I

D. Sivasubramanian, R. Iyer, G. Ramakrishnan, and A. De. Training Data Subset Selection for Regression with Controlled Generalization Error, June 2021.