ECE236A Project
Support Vector Machine, With Sample Selection

Mia-Sara Reyes Helene Levy Qotayba Alrayes
David Hill

December 8, 2021

Part 1
Compression of Data for Learning

1 Background and Objective

This project examines the significance of the number of data-points used when training a support-vector machine (SVM) classifier. A classifier is trained using vectors with associated labels. The objective is to predict the label of a vector given the entries of that vector. The accuracy of the classifier is determined by averaging the amount of correctly predicted labels with respect to the total amount. Here, we consider a classifier with a central node that receives constrained communication from distributed sensors and that will train a classifier using data-points obtained from those sensors. Yet, there is a cost, that the classifier will be attempting to minimize, associated with each data-point transmission. Some data-points will prove to be more "useful" than others when training the classifier.

2 Sample Selection Methodology

The sample-selection algorithm filters the full training data, selecting only points that are considered "useful" for training, forming a subset of reduced size. The goal is to reduce the amount of data needed for training, without sacrificing the accuracy of the classifier. Sample selection is complete when all the points in the full training data set are considered. The sample-selection operates as follows, for each point in the full training data:

1. If no data point has been selected yet, include the first point in the training subset.
2. If exactly one data point has been selected, include the next data point that has the opposite class label in the training subset. With these two data points, classifier parameters W and b are obtained as described in Section 3.
3. Each successive point, $(x_{\text{new}}, y_{\text{new}})$, is added to the training subset if the hinge loss of the point, $(1 - y_{\text{new}}(W^\top x_{\text{new}} - b))$ is greater than some threshold $t \in \mathbb{R}$. With the new training subset, classifier parameters W and b are then recalculated as described in Section 3.
4. If the hinge loss is less than the threshold is not included in the training subset.

3 Training Approach

For this project, two unique data sets are being used to to train the SVM. The user will be prompted to select one of them in advance. The first one is a Gaussian distributed data with means of the classes equal to $\mu = [-1, 1]$ and $\mu = [1, -1]$, and they will share the covariance matrix:

$$\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{3.1}$$

The second data set is downloaded from the MNIST database. It consists of handwritten 60000 training and 10000 testing example images. Given a data set, the obtained classifier is:

$$g(x) = W^\top x - b \tag{3.2}$$

Considering the labels (classes) $y_i = 1$ and $y_i = 7$, for example, the desired classifier satisfies the following:

- if $y_i = 1$: $W^\top x_i - b \leq -1$

- if $y_i = 7$: $W^\top x_i - b \geq 1$

The classifier parameters, W and b, are obtained by solving the following optimization problem with ℓ_1-regularization:

$$\text{minimize} \sum_{i=1}^{N_{\text{train}}} (1 - y_i(W^\top x_i - b)) + \lambda \|W\|_1 \tag{3.3}$$

Note that this problem is solved using Python's cxvpy package.

4 Testing and Results

The first performance evaluation was made by comparing the testing accuracy achieved through utilizing the full MNIST (using labels 1 and 7) and Gaussian training data sets to the sample-selected training datasets. From Table 4.1, the accuracy is maintained while drastically reducing the training size.

Table 4.1. Testing Accuracy

Training Data	Training Size	Testing Accuracy
Full MNIST	13007	99.4%
Selected MNIST	352	98.9%
Full Gaussian	12000	92.2%
Selected Gaussian	8079	92.2%

The amount of training-points which ensure the corresponding testing accuracies (50%, 65%, 80%, 95%) are shown in Table 4.2. These training sizes do not exactly achieve those percentages, yet, they represent the closest conservative result. For instance, an accuracy of 65% is not possible for the MNIST dataset as 2 data points are sufficient to exceed 70% in accuracy. Nonetheless, the sample-selected training data repeatedly reaches the specified accuracy with less data-points in comparison with the randomly-selected data. The correlation between accuracy and data magnitude is shown in Appendix B figure 0.1 (a) for the MNIST data, and in Appendix C figure 0.1 (a) for the Gaussian data.

The exact performance of the sample-selection procedure is compared to random sample selection in Appendix A Tables 0.1 and 0.2. The accuracies for random selection were generated by training the SVM on a random selection of the specified number of data points. The reported accuracies for random selection were averaged over 1000 realizations.

Table 4.2. Required training points for test accuracies of LP (Part 1)

	50%	65%	80%	95%
MNIST	1	2	5	7
Randomly Selected	1	2	7	33
Gaussian	1	10	23	–
Randomly Selected	1	5	44	–

Two parameters are optimized in this project: the regularization parameter, λ, and the threshold parameter in sample-selection, t. λ is the weight of the regularization term, which stabilizes the minimization and encourages sparsity in W. λ was optimized for the MNIST data using 50 trials, with results plotted in figure 0.1 (a). The threshold parameter t, for part I of the project on the other hand, is the value at which the hinge loss of a certain point becomes large enough to be included in the sampling-set. Representation of the performance of the SVM using different threshold values is seen in Appendix D figures 0.1 (b) for MNIST data and 0.1 (c) for Gaussian data. In addition, the pre-optimized parameters for threshold and λ are shown in figures 0.1 and 0.2 in Appendix D.

Part 2
Bounds of Data Compression

5 Background and Objective

In Part II of the project we again utilize an SVM binary classifier to classify the labels of the MNIST dataset and two-dimensional Gaussian distributed points. While the first part of the project performs sample selection in an online manner, this portion performs sample selection given the entire dataset. An integer linear program (ILP) is formulated to minimize the number of points used for training while maintaining the classification accuracy. The ILP will be solved through the equivalent linear program (LP) relaxation, by constructing an integral solution from the real solution.

6 ILP and LP Implementation

Methodology:
Our approach for generating the ILP was to focus on minimizing the distance between points of opposite class. First, we assigned an indicator variable, x_i, to every data point:

$$x_i = \begin{cases} 1, & \text{if image } X_i \text{ is chosen} \\ 0, & \text{otherwise} \end{cases}$$

Letting n_1 and n_2 denote the total number of data points in classes 1 and 2 respectively, the optimization problem solves for a vector $x \in \mathbb{R}^{n_1+n_2}$. The data matrix is assumed to take the form $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$, where X_1 corresponds to the n_1 rows of data points in class 1, and X_2 corresponds to the n_2 rows of data points in class 2.

To minimize the distance between points of opposite class, we generated a vector M containing sums of the norms of the differences between every data point and all the points of the opposite class. Accordingly, we define $M_1 \in \mathbb{R}^{n_1}$ as follows:

$$M_{1,i} = \sum_{j=1}^{n_2} \|X_{1,i} - X_{2,j}\|, \quad \text{for } i = 1, \cdots, n_1$$

Similarly, we define $M_2 \in \mathbb{R}^{n_2}$ as follows:

$$M_{2,i} = \sum_{j=1}^{n_1} \|X_{2,i} - X_{1,j}\|, \quad \text{for } i = 1, \cdots, n_2$$

Then, we let $M = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix}$. The objective is then to minimize $M^\top x$. However, the cvxpy solver is unable to handle the large entries of M associated with the MNIST data. Thus, when implementing the ILP, we instead minimize the objective function $\hat{M}^\top x$, where $\hat{M}$ is the normalized vector. To ensure that the optimal x is not $x = 0$, we define a threshold t. We then enforce the first n_1 values of x to sum to a value larger than this threshold. We also enforce this for the last n_2 values.

$$[\mathbf{1}, \mathbf{0}]x \geq t$$
$$[\mathbf{0}, \mathbf{1}]x \geq t$$

Additionally, we relax the integral constraint on the indicator variable by letting $0 \leq x_i \leq 1$, for $i = 1, \cdots, (n_1 + n_2)$.

The optimization problem is then:

$$\begin{array}{ll} \text{minimize} & \hat{M}^{\top} x \\ \text{s.t.} & \begin{bmatrix} -\mathbf{1} & \mathbf{0} \\ \mathbf{0} & -\mathbf{1} \\ \multicolumn{2}{c}{I} \\ \multicolumn{2}{c}{-I} \end{bmatrix} x \leq \begin{bmatrix} -t \\ -t \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix} \end{array}$$

The integral solution is then obtained by rounding the entries of the LP solution, x, to the nearest whole number.

7 Results

While the ILP successfully truncates the training size, it yields a lower testing accuracy in comparison to the online sample selection method in Section 4, and the full datasets (Table 7.1).

Table 7.1. Testing Accuracy

Training Data	Training Size	Testing Accuracy
Full MNIST	13007	99.4%
ILP Selected MNIST	4000	94.6%
Full Gaussian	12000	92.2%
ILP Selected Gaussian	2160	91.0%

The numbers of points required to reach each desired accuracy are listed in the following table and are presented graphically in Appendix B Figure 0.1 and Appendix C Figure 0.1. Again, for the ILP method, more training points are required to achieve the desired accuracies.

Table 7.2. Required training points for test accuracies of ILP (Part 2)

	50%	65%	80%	95%
MNIST	100	400	1000	4000
Gaussian	2112	2114	2116	–

Note that for the Gaussian data, the accuracy increases by 30% (from 50% to 80%) when the number of selected images increases by just 4. This is because the algorithm chooses data points whose summed distance to all the points of the opposite class is smallest, meaning it prioritizes data points of one class that appear within the data cloud of the opposite class. Only after enough points are chosen on the correct side of the classifier will the classification be correct, resulting in the sudden jump in accuracy.

In our ILP design, the only tunable parameter is the threshold t, corresponding to the enforced minimum number of selected points of each class. This number is linearly proportional to the number of selected images, so the classification accuracy as a function of this threshold exhibits the same behavior as the accuracy versus number of selected points. Namely, the accuracy undergoes a large jump as the threshold is increased marginally. This behavior is plotted for the Gaussian data in Appendix C Figure 0.1, in which the accuracy jumps from 50% to 80% as the threshold increases from 1056 to 1058. One potential remedy for such jumps would be to introduce an additional parameter to the design that ensures that the sum of the distances from each data point to all the points of the opposite class is above some number, thus minimizing the selection of outlying data appearing in the opposite point cloud. However, this issue does not occur for the MNIST data, so the current algorithm seems to be sufficient for realistic classification tasks.

Appendix A
Random Selection vs. LP Sample Selection

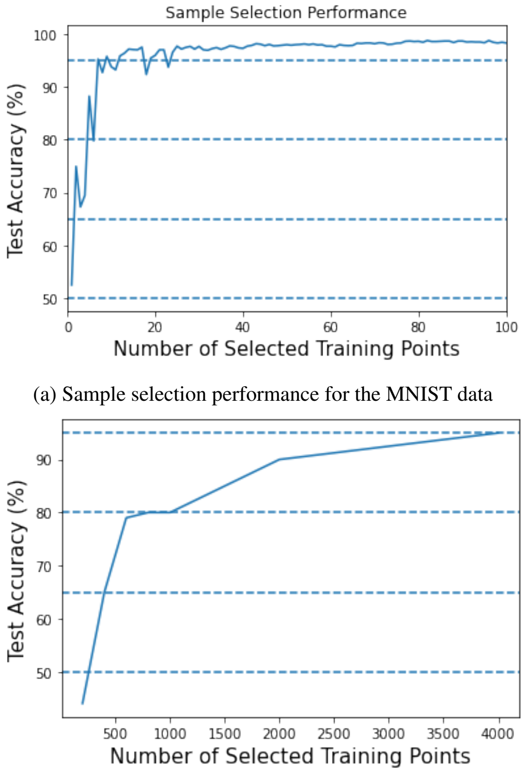
Table 0.1. Number of required data points and their corresponding percent accuracies for the Gaussian data of LP (Part 1)

Number of Data Points	1	10	23	7147
Sample Selection	50%	77.8%	81.5%	92%
Randomly Selected	63.5%	71.5 %	77.5%	92%

Table 0.2. Number of required data points and their corresponding percent accuracies for the MNIST data of LP (Part 1)

Number of Data Points	1	2	5	7
Sample Selection	52.4%	74.9%	88.3%	95.3%
Randomly Selected	59.2%	65.1%	78.9%	83.5%

Appendix B
Sample Selection Performance (MNIST Data)

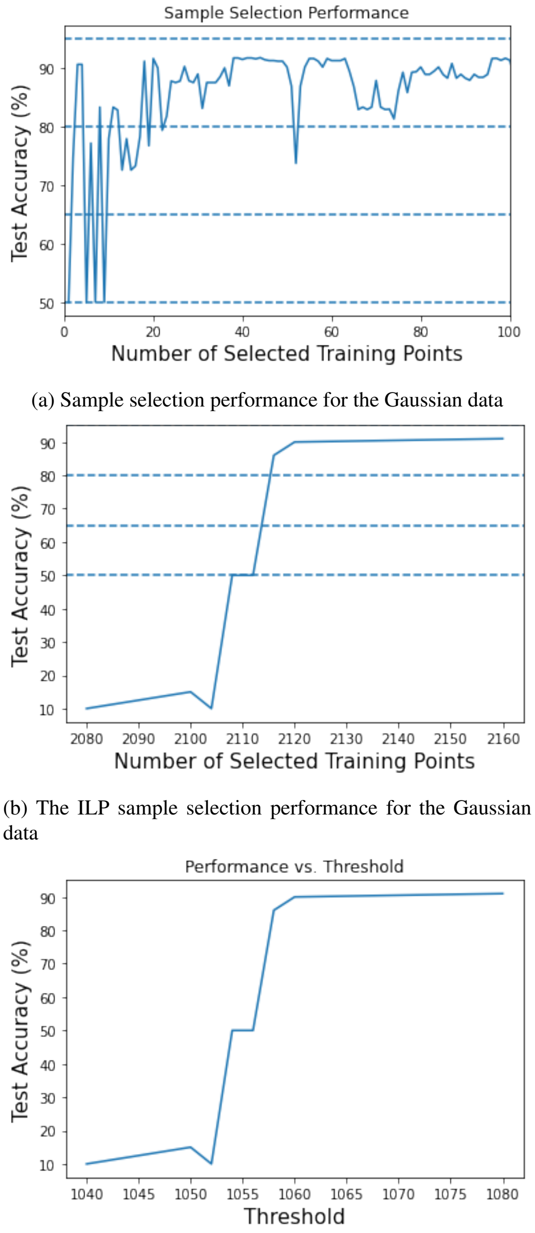


(a) Sample selection performance for the MNIST data

(b) The ILP sample selection performance for the MNIST data

Figure 0.1

Appendix C
Sample Selection Performance (Gaussian data)



(a) Sample selection performance for the Gaussian data

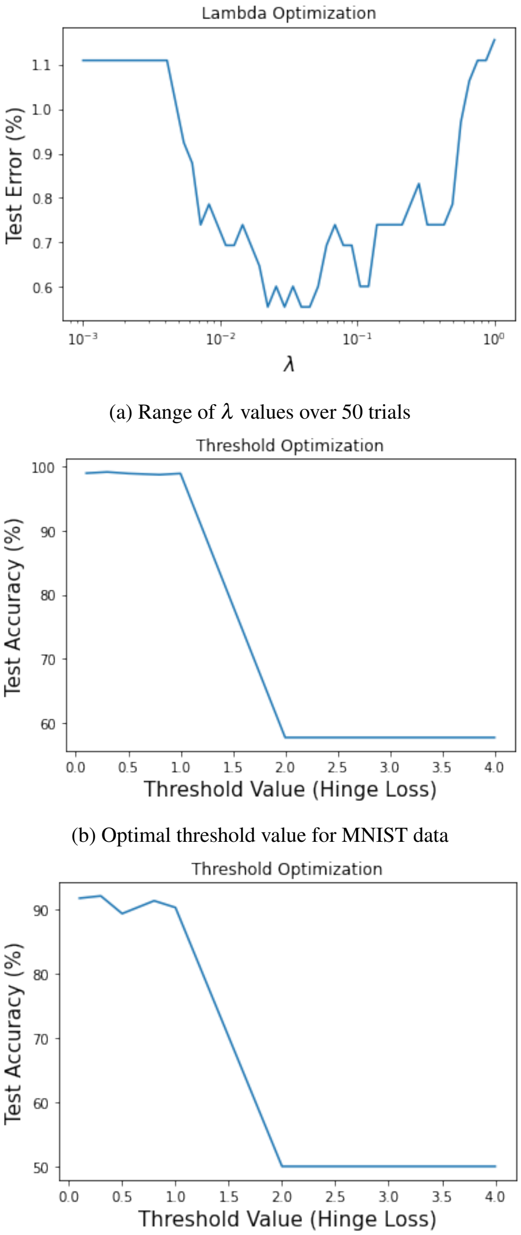
(b) The ILP sample selection performance for the Gaussian data

(c) The ILP accuracy versus threshold for the Gaussian data

Figure 0.1

Appendix D

Optimization Parameters



(a) Range of λ values over 50 trials

(b) Optimal threshold value for MNIST data

(c) Optimal threshold value for Gaussian data

Figure 0.1

Table 0.1. Optimal parameters for MNIST data

	Optimal Value
Regularization Parameter (λ)	0.03
Threshold Parameter (LP)	0.1
Threshold parameter (ILP)	2000

Table 0.2. Optimal parameters for Gaussian data

	Optimal Value
Regularization Parameter (λ)	0.001
Threshold Parameter (LP)	0.3
Threshold parameter (ILP)	1060