ECE236A Group6 Final Project

Group Members:
Yichen Yang, Jiusi Zheng, Wenjia Qi, Peian Xiao

December 1, 2023

1 Tasks

1.1 Task 1: Supervised Classification

1.1.1 Formulation

Suppose we have N data points in a training set, $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_N, y_N)\}$, where $\mathbf{x}_i \in \mathbb{R}^m$ and $y_i \in \mathbb{R}$ (m is the dimension of the data points). To find the best linear regression model, we can define the minimization problem

$$\min_{\mathbf{w},b} \sum_{i=1}^{N} |y_i - \mathbf{w}^T \mathbf{x}_i - b|$$

, where $\mathbf{w} \in \mathbb{R}^m$ and $b \in \mathbb{R}$ are the variables. To write the expression in a linear program, we define N variables, $t_1, t_2, ..., t_N$, where $t_i \in \mathbb{R}$. Then we can form the LP as follows:

$$\begin{array}{ll} \text{min} & \sum_{i=1}^{N} t_i \\ \text{subject to} & -t_i \leq y_i - \mathbf{w}^T \mathbf{x}_i - b \leq t_i, \quad i = 1, \ldots, n \end{array}$$

For the binary linear classifier, we define the decision function as $h(x) = sign(x)$, ($h(x) = 1$ if $x \geq 0$ else $h(x) = -1$). Together, the binary classifier is defined by

$$f(\mathbf{x}) = sign(\mathbf{w}^T \mathbf{x} + b)$$

Next, to construct a multi-class classifier, we adopt the one-vs-all classifier model (Mohamed, 2005). Suppose we have K classes of objects, first we train K binary classifiers for each pair of classes i, against the rest of the classes ($\{f_1(\mathbf{x}), ..., f_K(\mathbf{x})\}$). To predict the label of a given data point, we need to pass the data point through all the binary classifiers and then determine its class according to the largest value

$$\hat{f}(\mathbf{x}) = \text{argmax}_{i=1}^{K} f_i(x)$$

1.1.2 Implementation and Evaluation

For the synthetic data set, the highest accuracy we obtained is 0.958. For the MNIST data set, the highest we obtained is 0.963. The decision boundaries determined by the one-vs-all model for the synthetic data set are shown in Figure 1. In the process of finding the optimal hyperplane parameters, we also experimented with an alternative algorithm, which we named the Looping Traversal Algorithm. This method randomly generates hyperplane parameters within a certain range in the iteration. If the cumulative error was below a certain value, we considered these hyperplane parameters optimal. The decision boundaries determined by the algorithm are shown in Figure 2. Through comparison, we found that the gradient descent algorithm produced hyperplane parameters that resulted in a smaller classification error. Therefore, we ultimately adopted the optimal parameters generated by the gradient descent algorithm.

1.2 Task 2: Unsupervised Clustering

1.2.1 Formulation

The *big-M* term in this equation consists of a binary variable yj and the constant variable M (Cococcioni & Fiaschi, 2020). With this formulation, we can now model the constraints in the first equation as follows:

$$\mathbf{a_N} \leq b_N + M(1 - y_N)$$

$$\sum_{j=1}^{N} y_j = 1$$

$$y_j \in \{0,1\}, \text{for } j = 1, \ldots, N$$

The sum in the equation above ensures that only one of the N binary variables can assume the value one, that is only one of the constraints will be "activated". By using the big-M method mentioned above, we can form an ILP for Task 2. Each data point is assigned a binary variable b for every cluster

$$bi_j \in \{0,1\} \text{ for } i = 1, \ldots, n \text{ and } j = 1, \ldots, k.$$

A data set consists of di (i = 1, ..., n) data points which will be distributed among cj (j = 1, ..., k) clusters. The data points are assigned to the correct clusters by minimizing the sum of the distance between each data point and the cluster center. This distance is denoted r_i for i = 1, .., n. Then we can formulate the problem with n data points and k clusters as the following problem. For the problem below, we have tried the L1 and L2 norms. The accuracy of using norm 2 is better than the accuracy of using norm 1. According to the guidelines, the problem can be written as an integer linear problem as follows:

$$\min \sum_{i=1}^{n} ri$$

$$\text{s.t. } -t1_1 \leq d_1 - c_1 \leq t1_1$$

$$\mathbf{1}^T t1_1 \leq r_1 + M_1(1 - b\,1_1)$$

$$-t1_k \leq d_1 - c_k \leq t1_k$$

$$\mathbf{1}^T t1_k \leq r_1 + M_1(1 - b\,1_k)$$

$$-tn_k \leq d_n - c_k \leq tn_k$$

$$\mathbf{1}^T tn_k \leq r_n + M_n(1 - b\,n_k)$$

$$\sum_{j=1}^{k} b1_j = 1, \ldots, \sum_{j=1}^{k} bn_j = 1,$$

$$bi_j \in \{0,1\} \text{ for } i = 1, \ldots, n \text{ and } j = 1, \ldots, k.$$

1.2.2 Implementation and Evaluation

After experimenting with our clustering algorithm with different distances Figure 3, we found that we could achieve a higher performance with the L2 norm. For the synthetic data, the normalized mutual information (NMI) is 0.814, 0.832, 0.849 for cluster parameter K = 3, 5, 10, and the corresponding classification accuracy is 0.964, 0.972, 0.956. For the MNIST data set, the NMI is 0.300, 0.719, 0.862 for K = 3, 10, 32, and the corresponding classification accuracy is 0.498, 0.916, 0.952. Figure 4

1.3 Task 3: Semi-supervised learning

1.3.1 Formulation

In this task, we deployed an Integer Linear Program (ILP) to address the presented challenge. Our initial step involved the definition of variables and constants. Assuming that we know the data is of K clusters and we want to select the L most informative samples, our algorithm aims to cluster each data point and calculate the distance between each data point to the center of the cluster it belongs to. After that, we select $L/2$ furthest points and $L/2$ nearest points together to form the selected data list. The ILP is formulated as follows:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{N} t_i \cdot d_{1i} - \sum_{j=1}^{N} t_j \cdot d_{2j} \\
\text{s.t.} \quad & d_{1i},\ d_{2i} \in \{0,1\}, \quad i = 1, ..., N \\
& \sum_{i=1}^{N} d_{1i} = \frac{L}{2} \\
& \sum_{i=1}^{N} d_{2i} = \frac{L}{2} \\
& -\mathbf{1}^T t_i \leq \mathbf{x}_i - \mathbf{x}_{ci} \leq \mathbf{1}^T t_i
\end{aligned}$$

Here d_1 and d_2 are both indicators for the selection of data points. d_1 is for the data points with maximum distance to its center and d_2 is for the data points with minimum distance. Both the entries of the two indicators (vectors) need to add up to $L/2$. t is a slack variable indicating the L1 distance between data point x_i and x_{ci}, which is the center of the cluster that x_i belongs to. In addition, since the algorithm is related to the clustering algorithm that is implemented in Task 2, the chosen distance metric also influences the final result. Our group has tested both the L1 norm (shown in the LP) and the L2 norm. The L2 norm gave us a better performance. The outcome of the algorithm is shown in Figure 5.

1.3.2 Implementation and Evaluation

For both the synthetic data and MNIST data, we initialized our label selector with different cluster K values (same as the values in Task 2) and the outcomes are similar. Among all these algorithms, our algorithm and the random algorithm have relatively higher and more stable accuracy. The highest accuracy we achieved with the selected synthetic data is 0.962, 0.960, 0.968, 0.974, 0.972 for 5, 10, 20, 50, and 100 percent of the entire training set, while the accuracy for randomly selected data is 0.966, 0.956, 0.960, 0.974, 0.972. Meanwhile, the highest accuracy we achieved with the selected MNIST data is 0.830, 0.918, 0.912, 0.928, 0.934 for 5, 10, 20, 50, and 100 percent of the entire training set, while the accuracy for randomly selected data is 0.870, 0.892, 0.906, 0.918, 0.932 Figure 6. Although the difference between the performance is small, the relatively high accuracy does represent the strong capability of our label selection algorithm.

Except for the algorithm described before, we also came up with 5 more algorithms Figure 7. Firstly, the random algorithm chooses L data points randomly, then we find k center points of it. After this, we separate all data points into their clusters and calculate the sum of the distance between data points and their corresponding centers. Then we run the procedure thousands of times to find the minimum sum of distances so that we obtain the most informative samples. The rest 4 methods are similar. First and foremost, we obtain the center point of all data points. One method is to select samples that are closest to the center, the other one is to select the furthest samples. Moreover, we can choose half of the furthest points and half of the closest points. Lastly, we can select points that have a medium distance to the center.

2 Discussion and Comparison

1. Comparing the outcomes from Task 3 with those obtained in Task 1 reveals variations in the performance of our classifier, although within a relatively narrow range. The classifier for the MNIST dataset maintains an accuracy of approximately 0.85 with a mere 5% of the total training data, while the accuracy for the synthetic data is even closer to the highest accuracy we can achieve. In summary, while a larger sample size does influence the classifier's overall performance, the impact is modest when the existing data is sufficiently informative.

2. We assessed both clustering performance (NMI) and classification performance across varying training data sizes—specifically, 5%, 10%, and 50% of the provided dataset. The results demonstrated a decline in performance as the sample size decreased. For example, when randomly selecting 5% of the MNIST data with K set to 5, NMI decreased from 0.72 to 0.49. Conversely, when the sample size exceeded 20% of the original dataset, performance changes were minimal. In summary, reducing the sample size may not significantly impact the clustering algorithm's performance, except when the sample size becomes exceptionally small.

3. We can use this information in the design of the classifier. In designing clustering algorithms, leveraging soft decisions involves incorporating probability distributions or confidence scores for sample assignments to clusters, these algorithms output probabilities of membership to each cluster. Soft decisions from clustering can significantly benefit classifier design by incorporating these probabilities. Classifiers can weigh data points based on their certainty and this approach facilitates nuanced decision-making, allowing for adaptability to uncertain or ambiguous instances. The integration of soft decisions makes learning more efficient so even with a smaller number of samples we could achieve a better performance.

4. Semi-supervised learning is a type of machine learning where the model is trained on a dataset that contains both labeled and unlabeled examples. In the context of clustering, semi-supervised learning can be applied to assign class labels to clusters in a more unsupervised manner. This approach leverages a small amount of labeled data to extend class labels to unlabeled instances and then associates these labels with clusters. It allows for a more unsupervised assignment of class labels to clusters while benefiting from the limited labeled information available.

3 Reference

Mohamed, Aly (2005). "Survey on multiclass classification methods". *Technical Report, Caltech.*

Cococcioni, M., & Fiaschi, L. (2020). The Big-M method with the numerical infinite M. *Optimization Letters*, *15*(7), 2455–2468. https://doi.org/10.1007/s11590-020-01644-6

4 Appendix

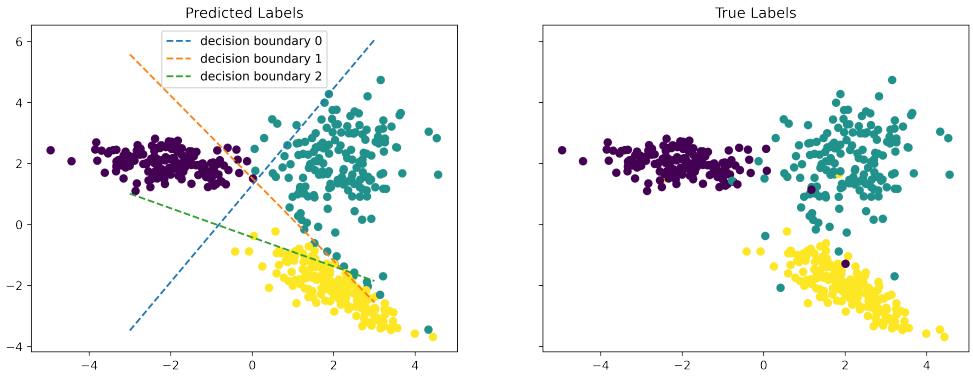
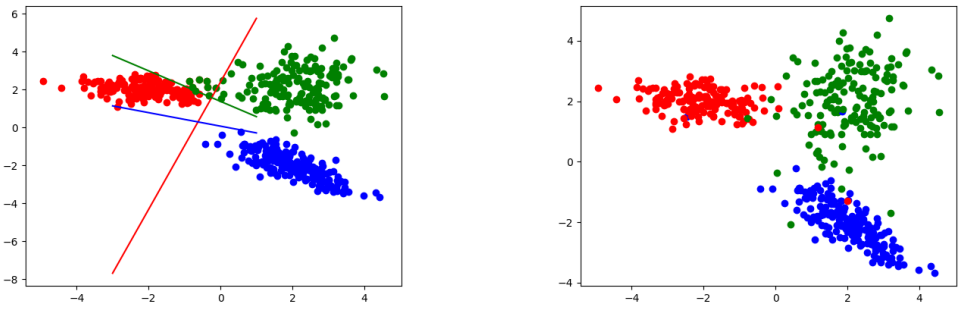


Figure 1: Task 1: Decision boundaries



(a) Predict label

(b) True label

Figure 2: Task 1: Decision boundaries with the looping traversal algorithm

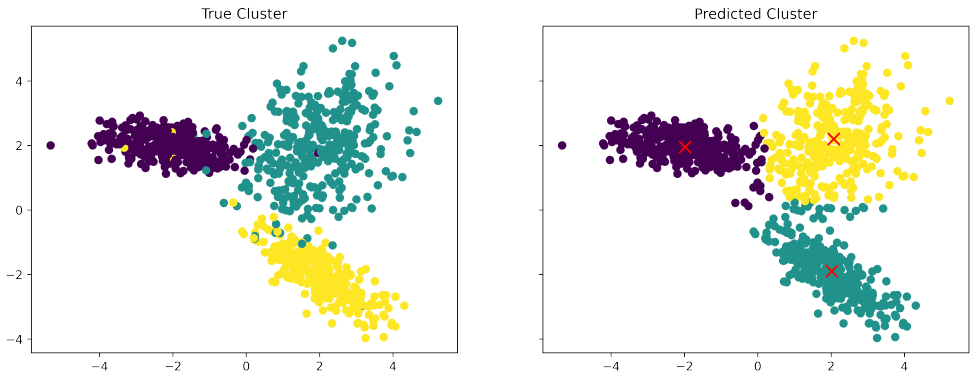
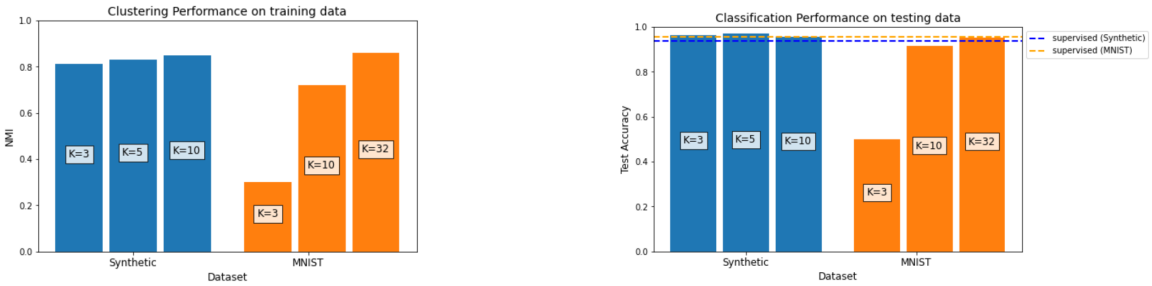


Figure 3: Task 2: Clustering algorithm: K=3



(a) Clustering performance

(b) Classification performance

Figure 4: Task 2: Performance analysis

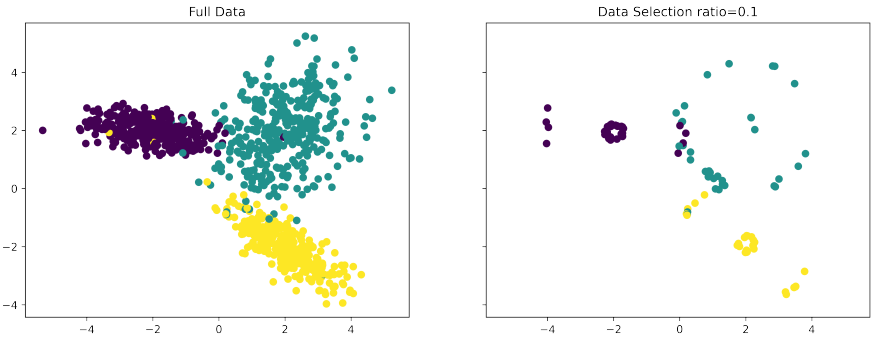
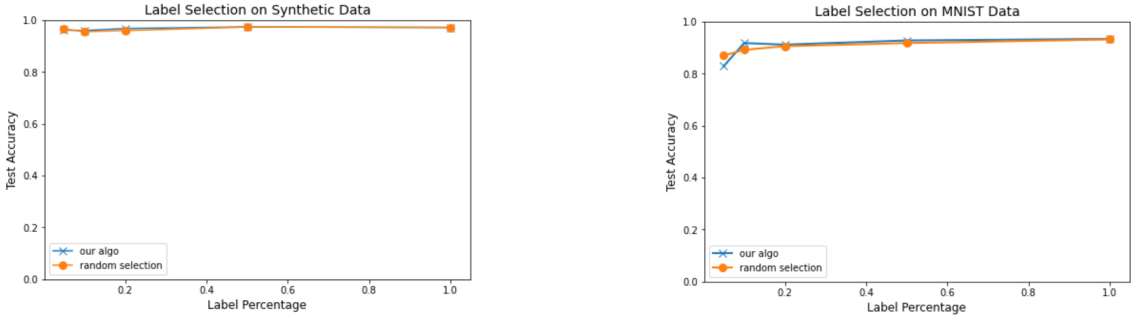


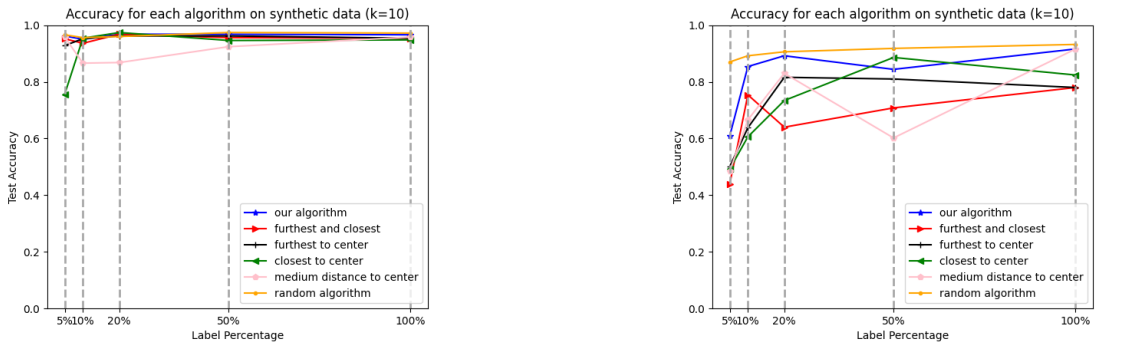
Figure 5: Task 3: Example for label selection (ratio=0.1)



(a) Label selection for synthetic data (k=3)

(b) Label selection for MNIST data (k=3)

Figure 6: Task 3: Performance of label selection algorithm on two data sets



(a) Label selection for synthetic data (k=10)

(b) Label selection for MNIST data (k=10)

Figure 7: Task 3: Performance comparison of different algorithms on two data sets