ECE236A Project Report **Fall 2023**

Linear Programming Project Report

Team member: Zhengtong Liu, Yiran Wang, Yunfan Li, Zijian Ding

1 Classification

1.1 One-to-one SVMs

We use one-to-one SVM to formulate the multi-class classification problem. We denote the size of the training set as N, the input dimension as n and the number of class as K. Specifically, for a dataset with K classes, we use $\frac{K(K-1)}{2}$ SVMs to classify between each pair of classes. During training, each SVM will be trained separately to maximize the margin between each pair of class. Note that we relax the penalty term from 2-norm to 1-norm in SVM to solve the problem by linear programing. During testing, we run all $\frac{K(K-1)}{2}$ SVMs and classify the data to the class with majority vote.

1.2 LP formulation

For each pair of class $(i, j), i < j$, suppose that we change class i to positive(+1) class and class j to negative(-1) class, we use the LP of the following form to get the hyperplane that separate i and j.

$$\min \sum_{i=1}^{N} \max\{0, 1 - y_i(\mathbf{a^T x_i} + b)\} + \lambda ||\mathbf{a}||_1$$

We further write the previous formulation in LP form:

$$\begin{aligned} \min_{u,t} \sum_{i=1}^{N} u_i + \sum_{i=1}^{n} t_i \\ s.t. \quad 0 \leq u_i, i = 1 \cdots N \\ \epsilon - y_i(\mathbf{a^T x_i} + b) \leq u_i, i = 1 \cdots N \\ a_i \leq t_i, a_i \leq -t_i, i = 1 \cdots n \end{aligned}$$

Denote the SVM for class $(i, j), i < j$ as $S_{i,j}$, and the solution as $\mathbf{a_{i,j}}$ and $b_{i,j}$.

During testing, for an input $\mathbf{x}$, we run $S_{i,j \neq i}$, for $\forall i, j$, and calculate the following:

$$c(\mathbf{x}, i) = \sum_{j>i} \mathbb{1}[\mathbf{a_{i,j}}^T \mathbf{x} + b_{i,j} \geq 0] + \sum_{j<i} \mathbb{1}[\mathbf{a_{i,j}}^T \mathbf{x} + b_{i,j} < 0]$$

Then the actual predicted class for input $\mathbf{x}$ is:

$$K(x) = \arg\max_i c(\mathbf{x}, i)$$

2 Clustering

2.1 KMeans++

We used the KMeans method to approach the multi-class clustering problem. Suppose we have N training samples and K clusters to identify. Our KMeans algorithm iteratively updates the predicted cluster label for each sample and the cluster centroids. In each iteration, we first assume the cluster centroids are known and use an LP to predict the data point labels; the centroid is then assigned to be the mean coordinate of each cluster. We initialize the cluster centroids in the same way as the KMeans++ algorithm.

2.2 LP formulation

Denote the data point as x_j ($j \in [N]$), the cluster centroids as c_k ($k \in [K]$), and the square of the distance between each data point and the closest centroid r_i ($i \in [N]$), we want to minimize the sum of the distance between each data point and the cluster centroid. We formulate the problem as below:

$$\begin{aligned}
&\min_{r} \sum_{i=1}^{N} r_i \\
&\text{s.t. } \|x_j - c_1\|^2 \le r_j \vee \|x_j - c_2\|^2 \le r_j \vee \cdots \|x_j - c_K\|^2 \le r_j \quad j \in [N]
\end{aligned}$$

We first convert the problem to an ILP: the introduced variables $b_j^k \in \{0, 1\}$ denotes whether data point j is in cluster k ($j \in [N]$ and $k \in [K]$); also denotes M_i ($i \in [N]$) a sufficiently large constant (say $M_i = \max_k\{\|x_i - c_k\|^2\}$). The ILP formulation is as follows:

$$\begin{aligned}
&\min_{r,b} \sum_{i=1}^{N} r_i \\
&\text{s.t. } \|x_j - c_k\|^2 \le r_j + M_j(1 - b_j^k) \quad j \in [N], \quad k \in [K] \\
&\quad \sum_{k=1}^{K} b_j^k = 1 \quad j \in [N] \\
&\quad b_j^k \in \{0, 1\} \quad j \in [N], \quad k \in [K]
\end{aligned}$$

If we relax it to an LP, we have the formulation:

$$\begin{aligned}
&\min_{r,b} \sum_{i=1}^{N} r_i \\
&\text{s.t. } \|x_j - c_k\|^2 \le r_j + M_j(1 - b_j^k) \quad j \in [N], \quad k \in [K] \\
&\quad \sum_{k=1}^{K} b_j^k = 1 \quad j \in [N] \\
&\quad b_j^k \ge 0 \quad j \in [N], \quad k \in [K]
\end{aligned}$$

Here is an intuition of our original ILP formulation: for fixed j, if $b_j^k = 1$, the constraint $\|x_j - c_k\|^2 \le r_j + M_j(1 - b_j^k)$ is reduced to $\|x_j - c_k\|^2 \le r_j$, which means the constraint is "activated"; otherwise, for $b_j^k = 0$, $\|x_j - c_k\|^2 \le r_j$ is not strictly imposed. This big-M constraint helps to recapitulate the "or" relationships.

3 Label Selection

3.1 Overall Algorithm

To select the labels, we first run a $K = 3$ clustering using the previous clustering LP formulation. Then, we use the pseudo label generated by the clustering to train three one-to-one SVM classifiers. For each classifier (i, j), we get the data points that belong to class i and j, and calculate their distance to the hyperplane. We then select the data points that are closest to the hyperplane (having the highest uncertainty according to pseudo labels). To balance the diversity of the selected data points, we choose an equal amount of data points for each one-to-one classifier (i, j), and an equal amount at both sides of the hyperplane.

3.2 LP formulation

We reuse the LP formulation in the previous two tasks. Specifically, we use the clustering formulation to get the pseudo labels and use the classifier's LP formulation to get the pseudo hyperplane.

4 Discussion and Comparison

1. In general, the classification accuracy increases as the training sample increases. The increase is more visible on the MNIST dataset and less obvious on the synthetic dataset. For each task, we re-train the model with $[1.0, 0.75, 0.5, 0.25, 0.1]$ of the dataset. We find that the supervised SVM can learn fairly well as long as the data ratio is above 0.5, the detail can be found in Figure 6. For the classification accuracy of the clustering algorithm, we note that the accuracy is almost 1 across different data size proportions for the synthetic dataset, while the accuracy increases from around 0.5 to above 0.8 for the MNIST dataset (Figure 7). Similar trends also exist for the classification accuracy after label selection, with an increase of 0.1 and 0.2 on the synthetic and the MNIST dataset when we increase the training samples (Figure 8). We conclude that a larger sample size might not be necessary and it depends on the dataset. For datasets like the synthetic one where the distributions are relatively simple and the classification task is relatively easy, we can get a good classification accuracy with a smaller dataset; in contrast, more training data can help to boost the classification accuracy on a complex dataset like the MNIST one.

2. As shown in Figure 7, the clustering NMI remains high (around 0.8) for the synthetic dataset while increasing from 0.25 to 0.7 on the MNIST dataset, as the number of samples ranges from 100 to 1000 samples (corresponds to 0.1 and 1 proportion of the training dataset). Therefore, the performance of clustering should generally increase with the number of samples, and the clustering algorithm might fail with too few samples. However, we caution that such a statement might also depend on the dataset. As we see from the results, the clustering algorithm performs well (NMI 0.8) even with 100 samples for the synthetic dataset. It might be that there are clear boundaries between each pair of clusters, so the clusters can be learned with a relatively small number of samples.

3. We can use the "soft decision" information to improve the accuracy of classification. Specifically, despite the limited data with labels, we can run clustering on labeled and unlabeled data, and use the labels of clustering with high certainty (in class * with prob * close to 1) to enrich the dataset. Those data should be close to the cluster centroids so the label generated by clustering should have high accuracy. And more data generally improves the accuracy of the classifier.

4. A way to assign class labels to clusters using a smaller number of true labels: we can select a few points that are close to the cluster centroids. Since we have high confidence that those samples belong to the specific clusters, we can use majority vote on the true labels of those selected samples to label the clusters.

References

[1] https://www.cvxpy.org/

[2] https://www.diva-portal.org/smash/get/diva2:1673547/FULLTEXT01.pdf

A Figures

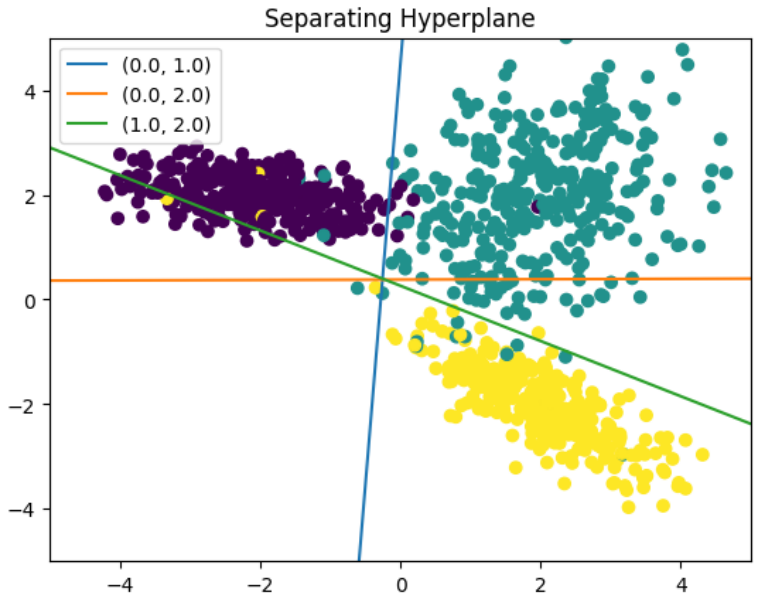

Figure 1: (Task 1) Separating hyperplane visualization on synthetic data

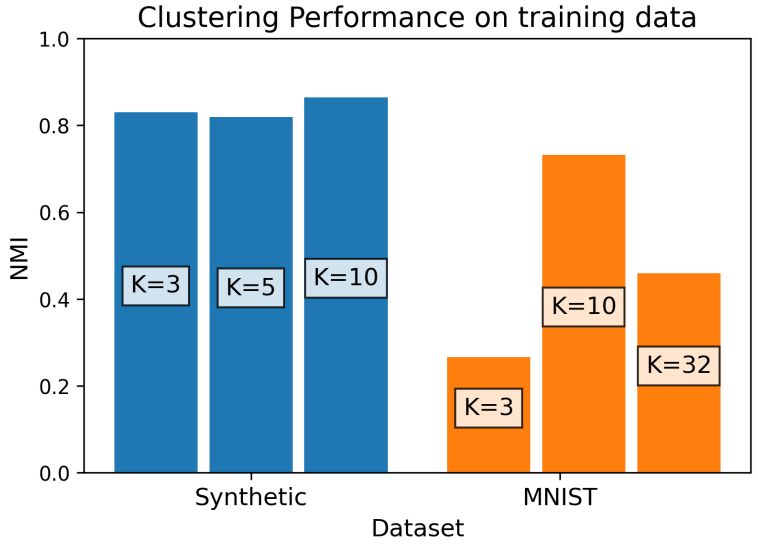

Figure 2: (Task 2) Normalized Mutual Information (NMI) on the training set

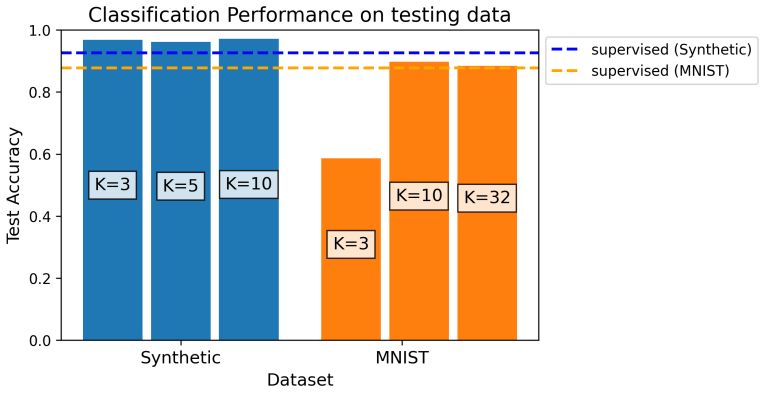

Figure 3: (Task 1 and 2) Comparison between classification accuracy on test set using Task 1 and 2

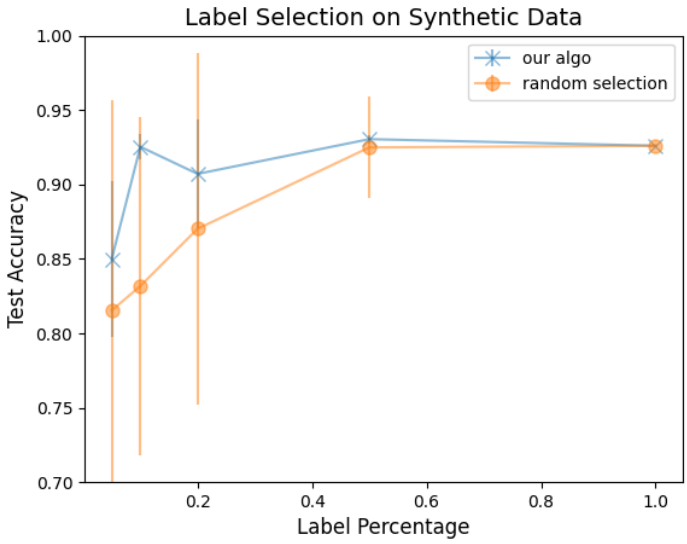

Figure 4: (Task 3) Comparison between the classification accuracy between random selection and our algorithm on the test set (synthetic data)

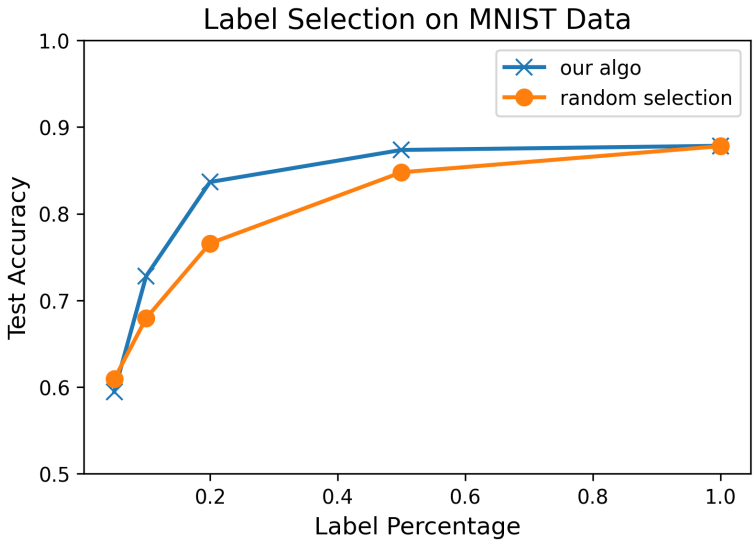

Figure 5: (Task 3) Comparison between the classification accuracy between random selection and our algorithm on the test set (MNIST data)

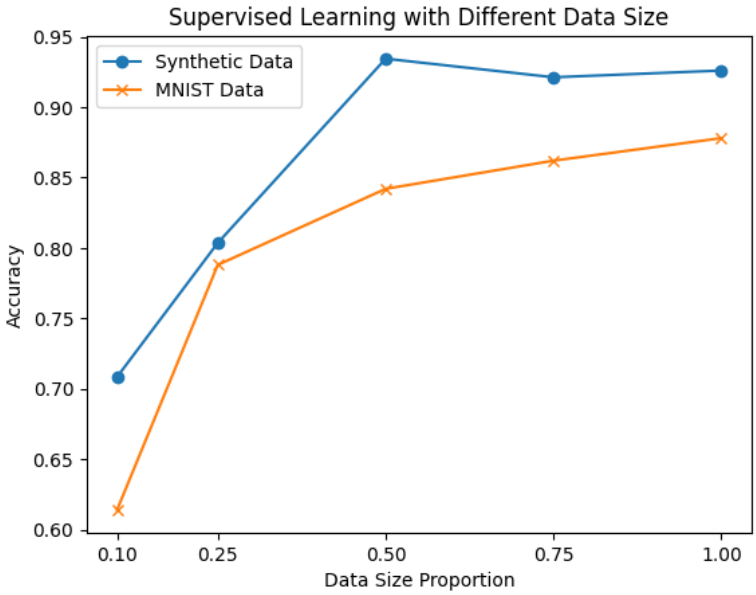

Figure 6: (Task 4.1) Comparison of Task 1 classification accuracy with different sizes of dataset.

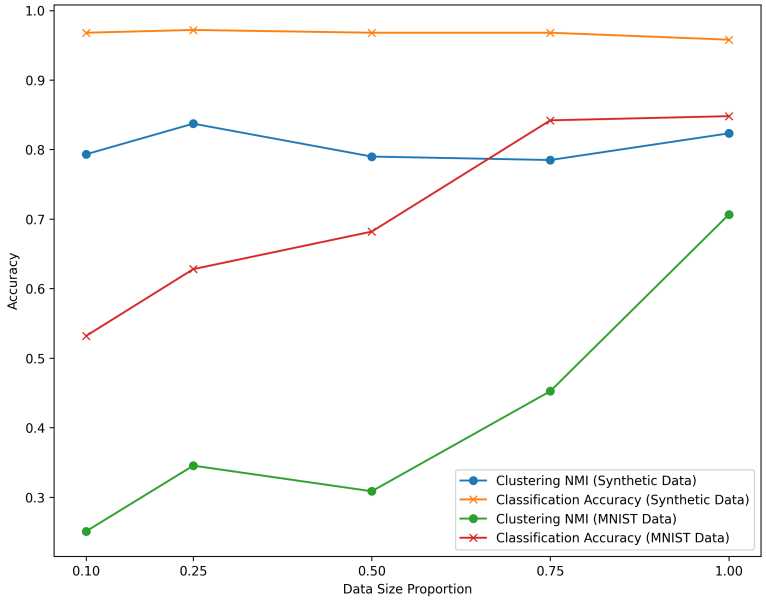

Figure 7: (Task 4.1 and 4.2) Comparison of Task 2 classification accuracy and clustering performance with different sizes of dataset. Here the number of clusters $K = 3$.

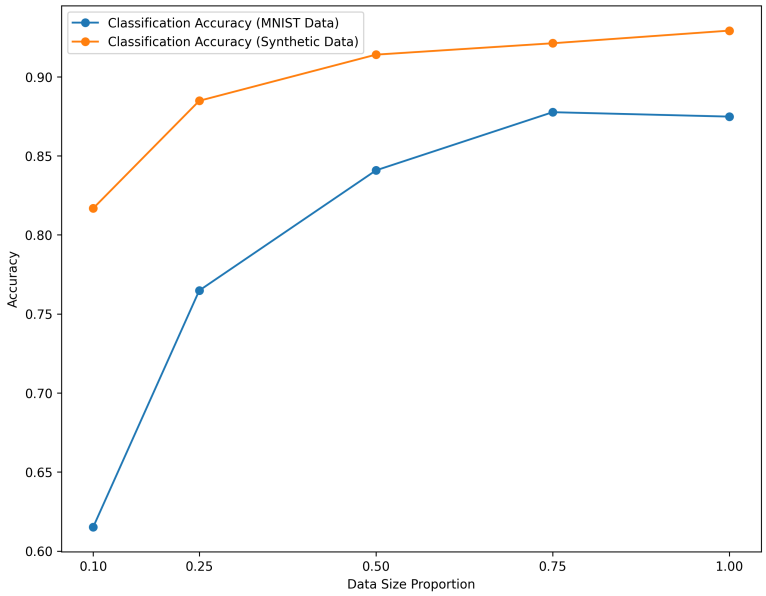

Figure 8: (Task 4.1) Comparison of Task 3 classification accuracy with different sizes of dataset. Here the proportion of data to obtain labels is 50%.