LP Formulation of Supervised and Unsupervised Classification Algorithms

Kartik Sharma, Mehmet Yigit Turali, Samyak Chakrabarty and Isaac-Neil Zanoria
Electrical and Computer Engineering, University of California, Los Angeles, CA, USA

November 28, 2024

1 Task 1: Supervised Classification

1.1 Variables

- $w_{jk} \in \mathbb{R}$: Weight of feature k for class j
- $b_j \in \mathbb{R}$: Bias term for class j
- $\xi_i \in \mathbb{R}_+$: Slack variable for point i

1.2 Classifier Creation Methodology

1. Optimization Objective: Solve a linear program to obtain:
   - Optimal weight vectors w_j and bias terms b_j for each class.
   - Slack variables ξ_i for handling misclassifications.
2. Decision Function: Create a classification rule defined as:

$$f(x) = \arg\max_j (w_j^T x + b_j)$$

3. Classification Mechanism:
   - Compute discriminant score for each class
   - Assign input to the class with the highest score
4. Practical Implementation: The classifier follows these core steps:

$$\begin{aligned}&\text{Training: Solve linear program to extract } (w_j, b_j)\\ &\text{Prediction: Compute } \max_j (w_j^T x + b_j) \text{ and classify.}\end{aligned}$$

1.3 Linear Problem for classification

$$\begin{aligned}
\text{minimize} \quad & \sum_{i=1}^{N} \xi_i + \lambda \sum_{j=1}^{K} \sum_{k=1}^{\tilde{M}} (u_{jk} + v_{jk}) & \\
\text{subject to} \quad & w_{y_i}^T \tilde{x}_i + b_{y_i} \geq w_j^T \tilde{x}_i + b_j + 1 - \xi_i & \forall i, j \neq y_i \\
& w_{jk} = u_{jk} - v_{jk} & \forall j, k \\
& u_{jk}, v_{jk}, \xi_i \geq 0 & \forall i, j, k
\end{aligned}$$

where λ is the regularization parameter and $\sum_{k=1}^{\tilde{M}} (u_{jk} + v_{jk})$ is for the L1 norm of w_j.

2 Task 2: Unsupervised Clustering

2.1 Variables

- $z_{ij} \in [0, 1]$: Soft assignment of point i to cluster j.
- $c_j \in \mathbb{R}^{\tilde{M}}$: Centroid of cluster j in reduced feature space.

2.2 Clustering Creation Methodology

The algorithm was designed to achieve efficient clustering by addressing key challenges in high-dimensional data. This methodology leverages dimensionality reduction, probabilistic initialization, and a relaxation-based clustering framework.

- **Dimensionality Reduction:** High-dimensional data often contain redundant or irrelevant features, which can negatively impact clustering performance. To address this we are using PCA while preserving $\alpha = 95\%$ of the variance.
- **Centroid Initialization Using K-means++:** To ensure robust clustering, K-means++ initialization was chosen for its ability to place centroids probabilistically, maximizing initial cluster separation which avoids poor local minima commonly encountered in standard K-means.

2.3 Formulate and Solve LP

To handle soft assignments:

- The clustering task was modeled as an Integer Linear Programming (ILP) problem, but to simplify computation, the ILP was relaxed to a Linear Programming (LP) problem.

Original ILP Formulation:

$$\begin{aligned} \text{minimize} \quad & \sum_{i=1}^{N}\sum_{j=1}^{K} z_{ij}\|x_i - c_j\| \\ \text{subject to} \quad & \sum_{j=1}^{K} z_{ij} = 1 \quad \forall i \\ & z_{ij} \in \{0,1\} \quad \forall i, j \end{aligned}$$

Relaxation to LP:

$$0 \leq z_{ij} \leq 1 \quad \forall i, j$$

2.3.1 Step 4: Centroid Update

Soft assignments provide weights for each point's contribution to a centroid:

$$c_j = \frac{\sum_{i=1}^{N} z_{ij} x_i}{\sum_{i=1}^{N} z_{ij}}$$

This step ensures centroids move towards the weighted mean of assigned points, improving cluster consistency.

3 Task 3: Feature Efficient Learning

3.1 Variables

- $s \in \mathbb{R}^M, s_i \in [0,1]$: Feature selection vector (N total training samples)
- $t_{\text{min}} \in \mathbb{R}$: Auxiliary variable for the minimum "sum absolute covariance" across all class pairs (**supervised**)
- $t \in \mathbb{R}$: Sum of selected pixel variances (**unsupervised**)
- $d_{\text{cols}} \in \mathbb{R}$: regularization variable 1, representing column sum dispersion of the feature mask
- $d_{\text{rows}} \in \mathbb{R}$: regularization variable 2, representing row sum dispersion of the feature mask

3.2 Feature Selection Methodology

Our proposed feature selector is constructed through the following approach:

1. Feature value metric **(Supervised Learning)**:
   - **Goal:** Find pixel indices that have the *highest absolute covariance with the training sample labels.*
   - **Procedure:** For each pairing of class labels, take the training samples with those 2 labels, and standardize by renaming labels to (+1, -1), and compute the covariance between the value of that pixel across the reduced training samples with renamed labels. Covariance map for each pair of classes can be seen in 7.
2. Feature value metric **(Unsupervised Learning)**:
   - **Goal:** Find pixel indices that have the highest variance across the training set, which we assume are most useful for unsupervised classification or clustering. Variance map for the dataset can be seen in Fig. 8

   Features selected by supervised and unsupervised methods shown in Fig. 6
3. Regularization (**Supervised** and **Unsupervised**):
   - **Goal:** Add a regularization term that incentivizes spatial diversity in the selected features. This is to avoid picking highly variant/covariant but close (so highly correlated) pixels.
   - **Procedure:** For features selection vector, calculate column and row sums:

     $$\sigma_{cols,i} = \Sigma_{j=1}^{\sqrt{M}} s_{(i+j\sqrt{M})}, \quad 1 \leq i \leq \sqrt{M}$$
     $$\sigma_{rows,i} = \Sigma_{j=1}^{\sqrt{M}} s_{(i\sqrt{M}+j)}, \quad 1 \leq i \leq \sqrt{M}$$

     where $\sqrt{M}$ is the side length of the sample image (assume square samples).
   - Actual variance calculation requires $(\cdot)^2$, which is not a linear constraint, so use the following analog for the dispersion metric:

     $$d_{\text{cols}} = \frac{\Sigma|\sigma_{\text{cols}} - \overline{\sigma}_{\text{cols}}|}{K}$$
     $$d_{\text{rows}} = \frac{\Sigma|\sigma_{\text{rows}} - \overline{\sigma}_{\text{rows}}|}{K}$$

   A demonstration of features selected by just the regularization term can be seen in Fig. 11

3.3 Formulating the LP

3.3.1 Optimization Problem (Supervised Learning):

- Formulate as a *maxi-min problem* - maximize the minimum (over the possible class pairings) covariance sum of the selected pixels.
- Attach the column/row dispersion regularization terms (with hyperparameter coefficients λ_1, λ_2) to increase the spatial diversity of the features.
- Relax $s_i \in \{0, 1\}$ binary pixel constraints to $0 \leq s_i \leq 1$ for computational tractability. Select N highest weight features at the end of training.

Original ILP:

$$\begin{aligned} \underset{s}{\text{maximize}} \quad & \min(t_{(1,2)}, t_{(1,3)}, \dots, t_{(K-1,K)}) \\ \text{subject to} \quad & t_{(i,j)} \leq \Sigma_{i=k}^{M}(s_k u_k^{(i,j)}) \ \forall\, i, j \in \text{classes}, i \neq j \\ & 1^T s \leq N \\ & s \in \{0, 1\} \end{aligned}$$

LP Relaxation with regularization:

$$\begin{aligned}
\underset{s}{\text{maximize}} \quad & t_{min} - \lambda_1 d_{cols} - \lambda_2 d_{rows} \\
\text{subject to} \quad & t_{min} \leq s^T u_{(i,j)} \; \forall \; i, j \in \text{classes}, i \neq j \\
& 1^T s \leq N \\
& \sigma_{cols,i} = \Sigma_{j=1}^{\sqrt{M}} s_{(i+j\sqrt{M})}, \quad 1 \leq i \leq \sqrt{M} \\
& \sigma_{rows,i} = \Sigma_{j=1}^{\sqrt{M}} s_{(i\sqrt{M}+j)}, \quad 1 \leq i \leq \sqrt{M} \\
& d_{cols} = \frac{\Sigma|\sigma_{cols} - \overline{\sigma}_{cols}|}{K} \\
& d_{rows} = \frac{\Sigma|\sigma_{rows} - \overline{\sigma}_{rows}|}{K} \\
& 0 \leq s \leq 1
\end{aligned}$$

3.3.2 Optimization Problem (Unsupervised Learning):

- Formulate as a *maximization problem* - maximize the sum variance of the selected pixels.
- Same regularization terms and relaxation as supervised learning.

Original ILP:

$$\begin{aligned}
\underset{s}{\text{maximize}} \quad & t \\
\text{subject to} \quad & t \leq s^T v \\
& 1^T s \leq N \\
& s \in \{0, 1\}
\end{aligned}$$

LP Relaxation with regularization:

$$\begin{aligned}
\underset{s}{\text{maximize}} \quad & t - \lambda_1 d_{\text{cols}} - \lambda_2 d_{\text{rows}} \\
\text{subject to} \quad & t \leq s^T v \\
& 1^T s \leq N \\
& \sigma_{\text{cols},i} = \Sigma_{j=1}^{\sqrt{M}} s_{(i+j\sqrt{M})}, \quad 1 \leq i \leq \sqrt{M} \\
& \sigma_{\text{rows},i} = \Sigma_{j=1}^{\sqrt{M}} s_{(i\sqrt{M}+j)}, \quad 1 \leq i \leq \sqrt{M} \\
& d_{\text{cols}} = \frac{\Sigma|\sigma_{\text{cols}} - \overline{\sigma}_{\text{cols}}|}{K} \\
& d_{\text{rows}} = \frac{\Sigma|\sigma_{\text{rows}} - \overline{\sigma}_{rows}|}{K} \\
& 0 \leq s \leq 1
\end{aligned}$$

Note: We know absolute values and sums can be rephrased as linear constraints with added auxiliary variables. This form is kept for readability.

4 Discussion and Comparison

4.1 Supervised vs Unsupervised Learning

- The unsupervised learning algorithm (here K-means) depends on cluster density and hence requires more samples to perform accurate clustering. However since it assumes clusters to be convex spheres, it can struggle with non-convex clusters. For example, two elongated clusters that overlap are difficult to be seperated by K-means clustering but easier using SVMs.
- Figure 1 shows SVM algorithm performing better at lower sample sizes on MNIST. The gap reduces for larger samples, but overall performance of SVM is better than simple K-means.

4.2 Clustering performance and Sample size

- Accuracy (Figure 1) increases with sample size, as expected.
- If the sample space is too small, NMI can still be high as seen in Figure 2. This is because the variability in the clusters are low owing to less number of points. The labels are misaligned, owing to low accuracy, even though the clusters appear more consistent.

4.3 Soft Decisions Classifier

- If clustering were to give soft decisions, it is effectively packing more information in the output. This could potentially help with better label alignment. Figure 14 supports our hypothesis.
- If decisions are soft, instead of using majority vote to align labels, one could use highest total score for each label in a cluster. Figure 4 shows that this performs fairly well.
- Another way of using the score is to add it as a feature to the data points to train an SVM after clustering.
- As seen in Figure 14, Soft decision based K-means can give higher accuracy with much less samples ($> 20\%$) compared to hard ($> 50\%$).

4.4 Class Label Alignment

- A better way to assign class labels can be using labels for only 10% of the points that are closest to the centroids in each clusters. These points can be considered representative points since they are most likely to be correct. Figure 5 confirms our hypothesis.

5 Key Observations

5.1 Supervised vs. Unsupervised Learning

- **Supervised Learning Performance:**
  - Demonstrated higher accuracy and more reliable classification results
- **Unsupervised Clustering Characteristics:**
  - Performance heavily influenced by the number of clusters K.

5.2 Impact of Constraints in Task 3

- **Data Efficiency Trade-offs:**
  - Reduction in labeled samples or features revealed critical performance trade-offs
  - Demonstrated sensitivity of model performance to data constraints
- **Feature Selection Insights:**
  - Careful selection of informative features maintained performance comparable to full-feature models.

5.3 Challenges and Insights

- Demonstrated the significant advantage of labeled data
- Revealed fundamental constraints of clustering algorithms
- Exposed challenges in unsupervised learning, particularly with complex or non-trivial data distributions
- Provided a critical bridge between supervised and unsupervised methods
- Demonstrated that strategic labeling and feature selection can optimize performance
- Showed potential for resource-efficient machine learning strategies

6 Task 1: Decision Boundaries

The synthetic dataset had only two features so a 2D decision boundary was plotted to visualize the performance of the classification LP (Fig. 12). However, the Fashion MNIST dataset is made of 784 features (or pixels) we can't plot a decision boundary directly but we can do a PCA first to reduce the dimensionality of the dataset from 784 to 2 (though it won't be a good classifier, still it is possible to do that) and then we can plot the decision boundary (Fig 13).

7 Task 3 Results

7.1 Selected Features

The feature masks selected using the linear programs from Section 3 for each "max features" constraint are shown in Fig. 6. Qualitative observations about the features masks are recorded here:

- Especially in the supervised learning case, we see the selector returns features that satisfy intuitively "how would a human differentiate between a shirt, coat, and dress?" For the supervised case, with features = 80, 320, we can recognize the outline of a garment. **The selected features mostly lying on the "sleeves"** of the image makes sense - they are the main differentiator between a dress, t-shirt, and coat.
- The spatial regularization term seems to function as intended. Especially in the unsupervised learned masks, the features selected are distributed evenly along the x and y directions.

7.2 Performance vs Random Selection

The performance of the models from Tasks 1 and 2, when trained with Task 3 selected features vs random features, are recorded in Fig. 9 and Fig. 10 respectively.
Comments on the performance differences with selected versus random features are below:

- In the case where the max feature constraint is less than the original number of features, we see the models trained with selected features outperform those trained with random features in all cases but the supervised, $M = 320$ case, which we attribute to randomness.
- We see the performance advantage of the selected features over the random features is highest in for lower M values, which is what we would expect - the probability of the IID mask containing a highly-predictive pixel is lower with smaller M.
- As M increases, the advantage from feature selection weakens. As mentioned in Section III, while trying to maximize variance or covariance our optimization neglects whether selected pixels are correlated with each other. The spatial diversity of the IID feature mask means it finds more uncorrelated pixels at higher M values and can perform closer to the selected pixels.

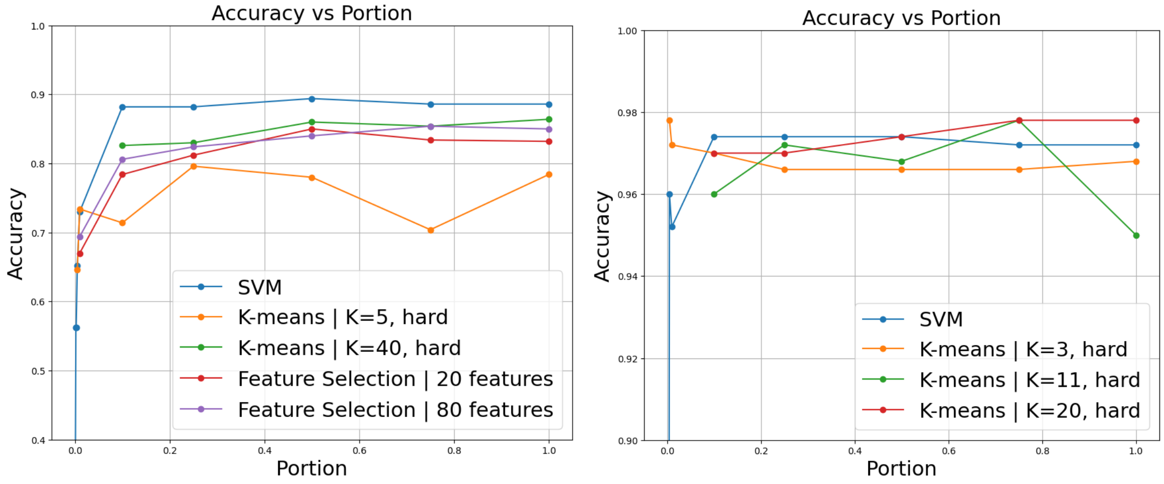

Figure 1: Accuracy vs Sample size for all 3 Tasks in a) Fashion MNIST data, b) Synthetic Data

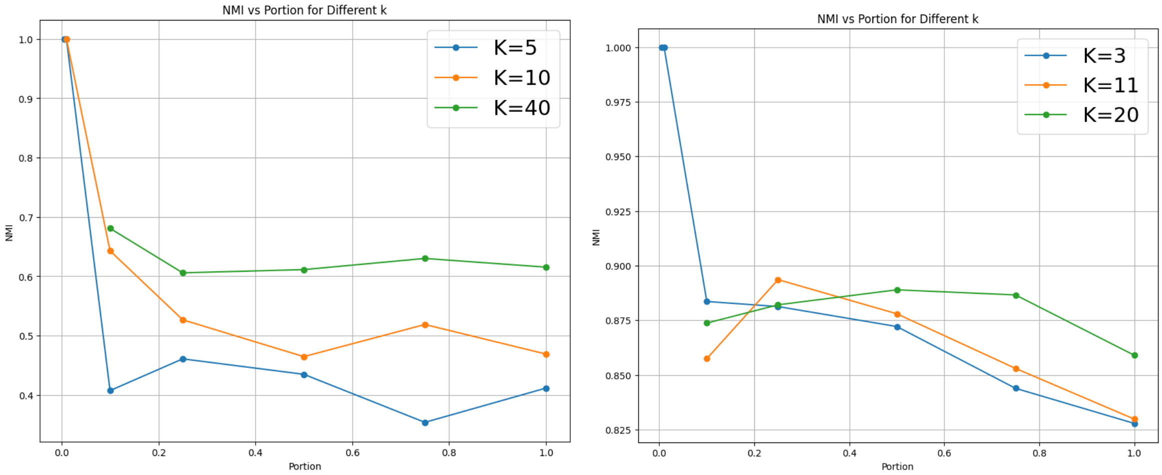

Figure 2: NMI vs Sample size for Clustering in a) Fashion MNIST data, b) Synthetic Data

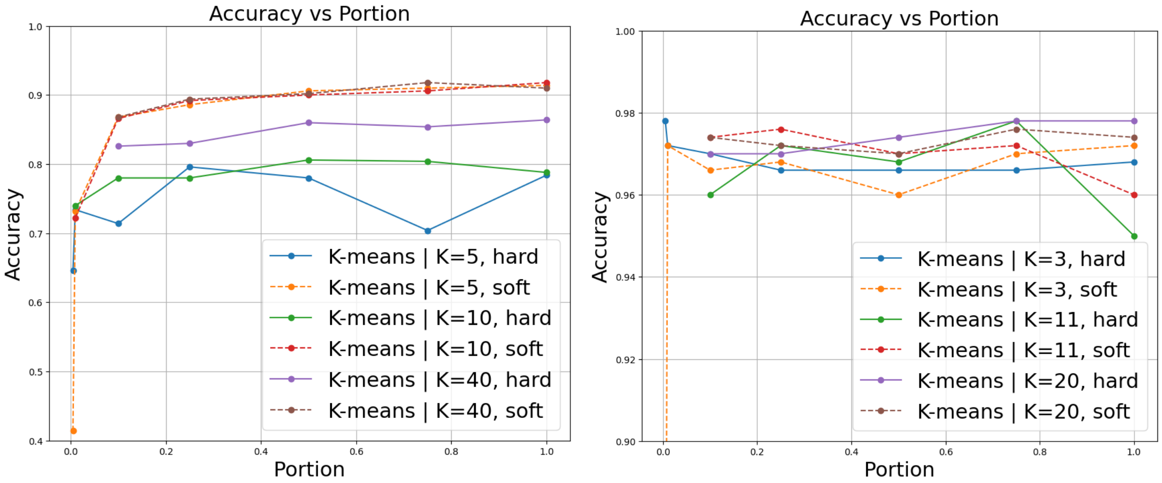

Figure 3: Hard vs Soft Clustering in a) Fashion MNIST data, b) Synthetic Data

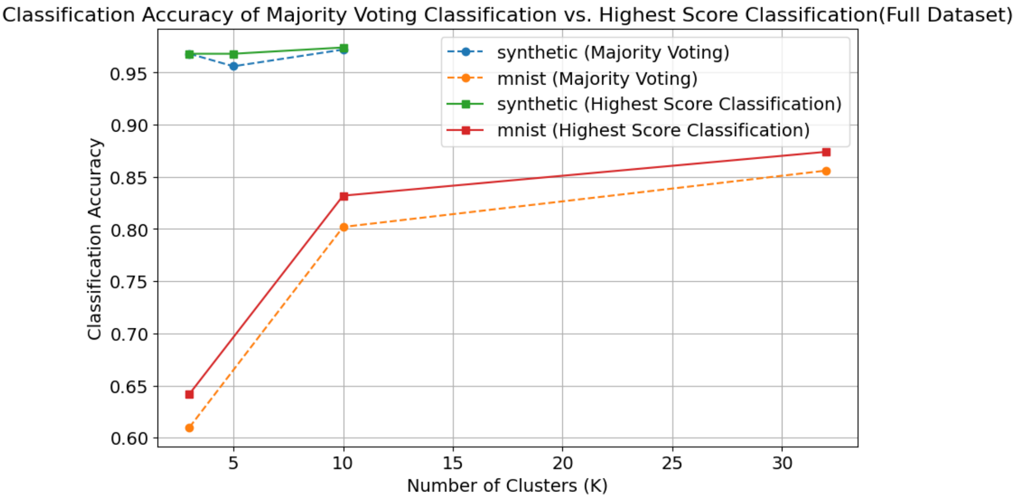

Figure 4: Majority Voting vs Proposed Scheme for Soft Clustering based Classifier

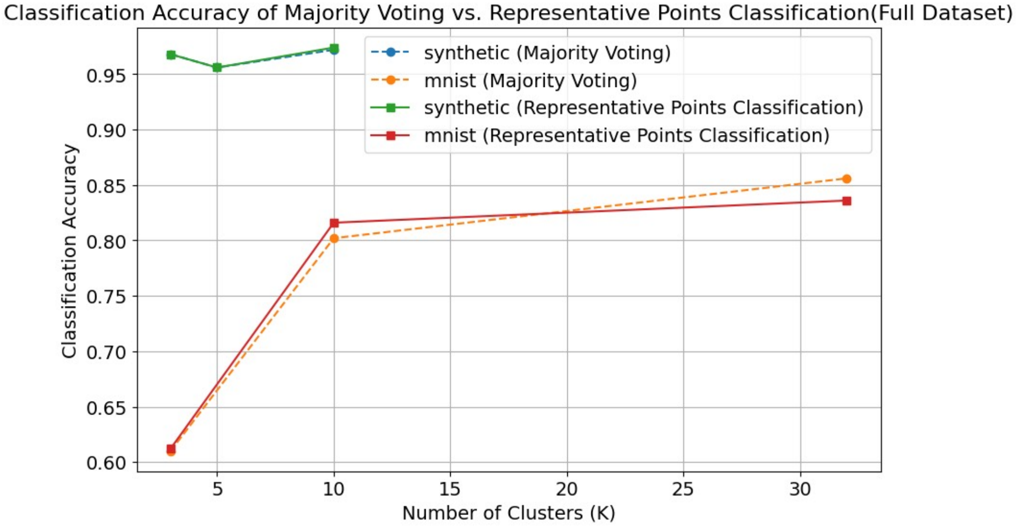

Figure 5: Proposed Labeling Scheme for Clustering

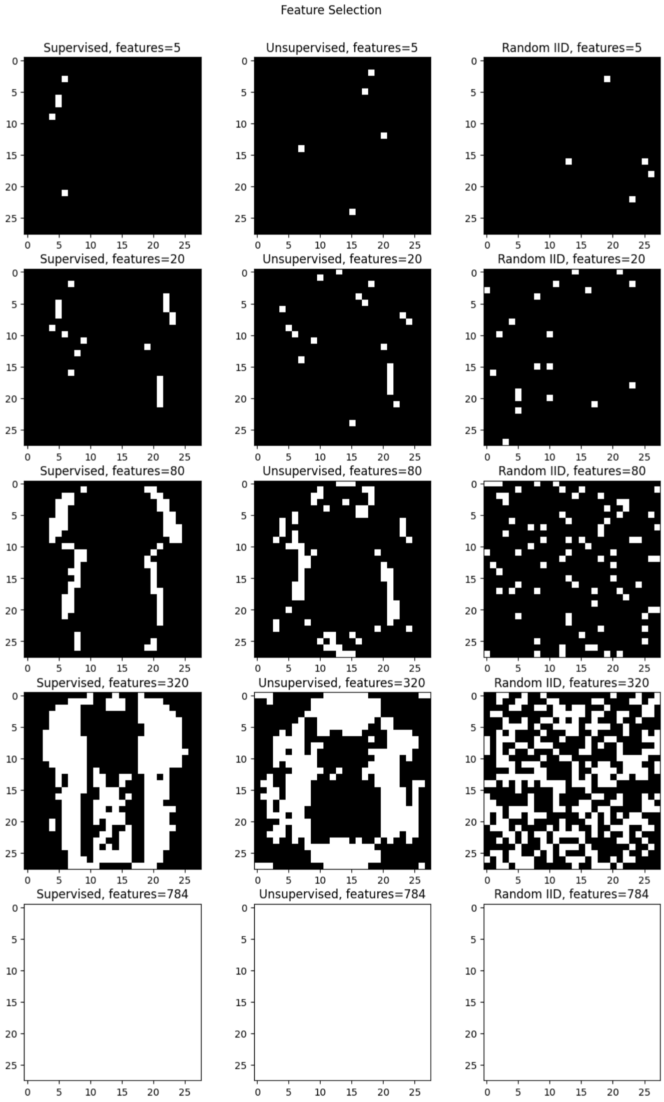

Figure 6: The feature masks produced by (left to right) Supervised, Unsupervised, and Random Feature Selection, for (top to bottom) [5, 20, 80, 320, 784] max features

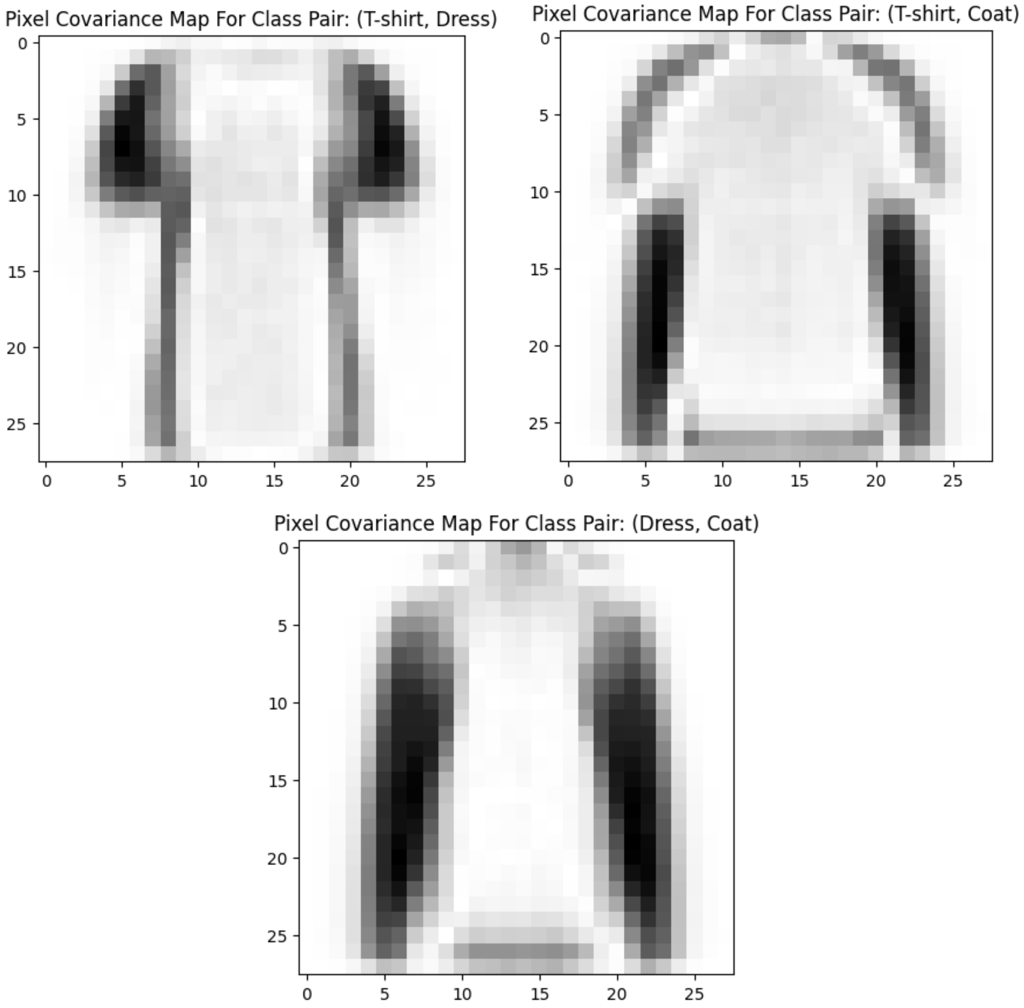

Figure 7: Pixel covariance maps for each class pair in the dataset, used for supervised feature selection in Task 3.

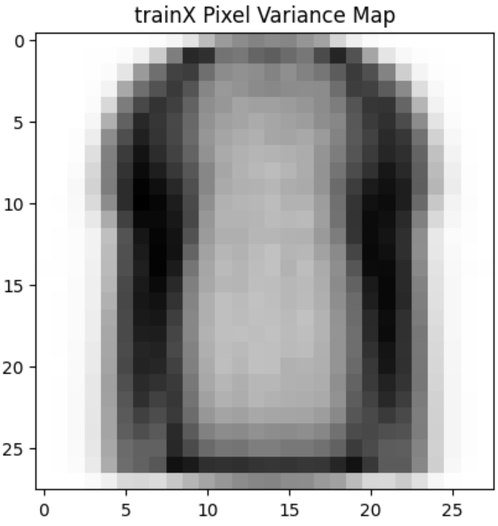

Figure 8: Pixel variance map, used for unsupervised feature selection in Task 3.

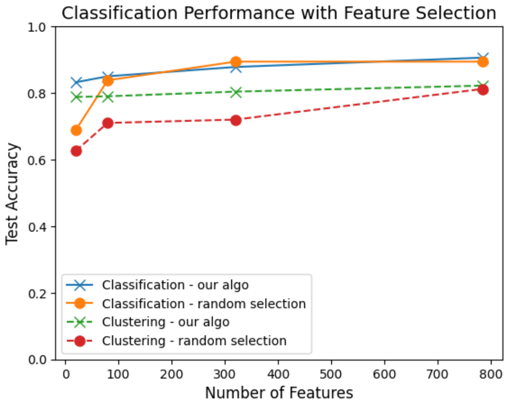

Figure 9: Classification performance using Task 3 selected features.

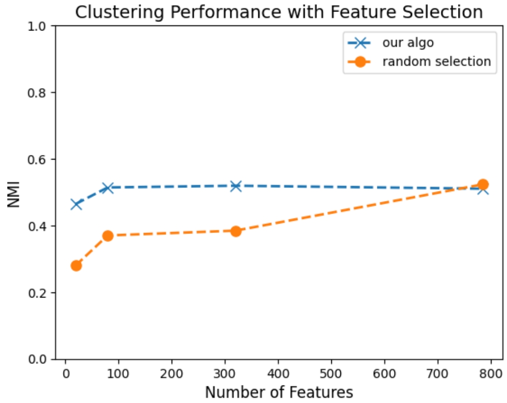

Figure 10: Clustering performance using Task 3 selected features.

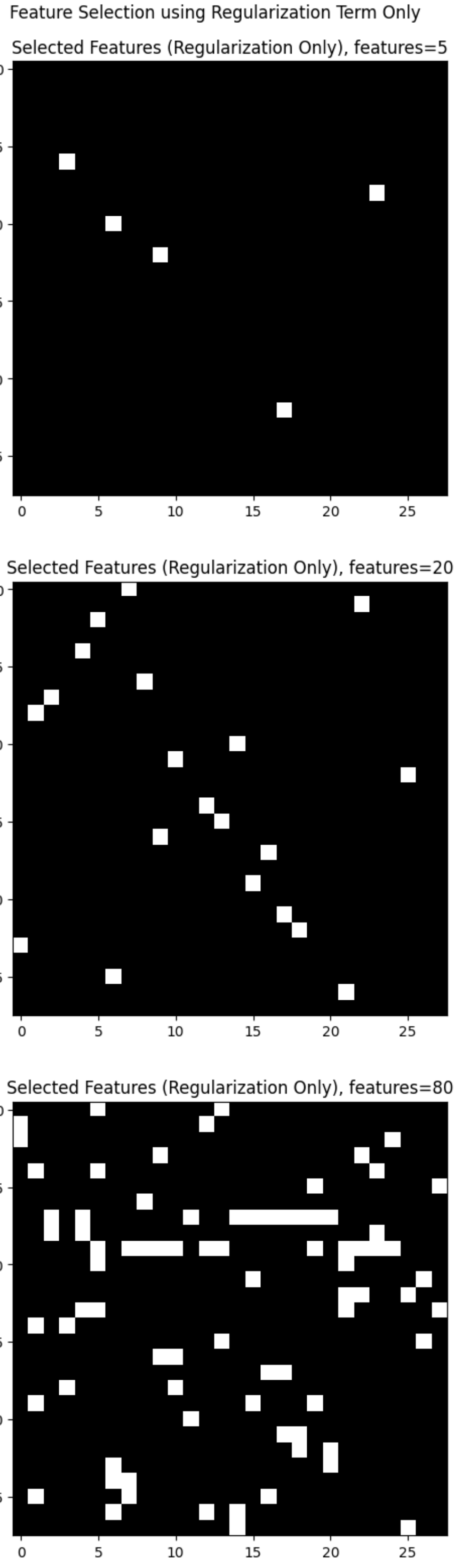

Figure 11: Task 3 feature selection using regularization (increase spatial dispersion) term only.

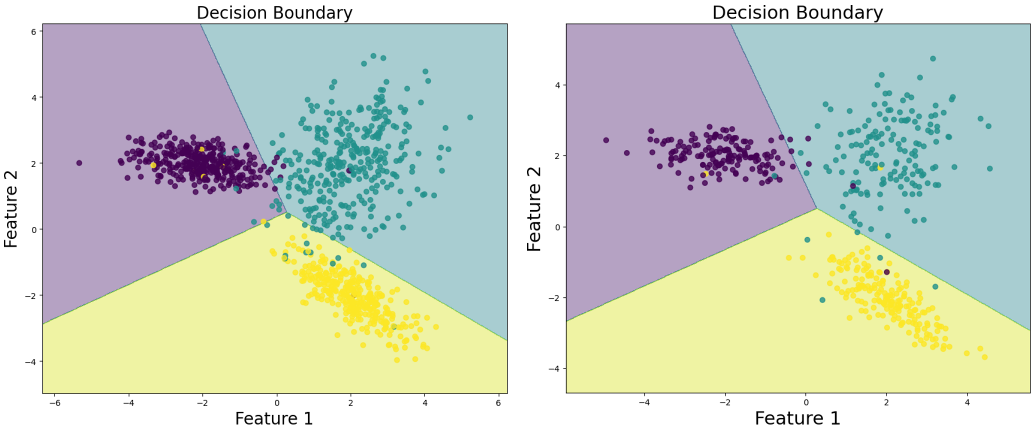

Figure 12: (a) Inference on the training data to check if there is any overfitting. (b) Plotting the decision boundary for the test dataset.

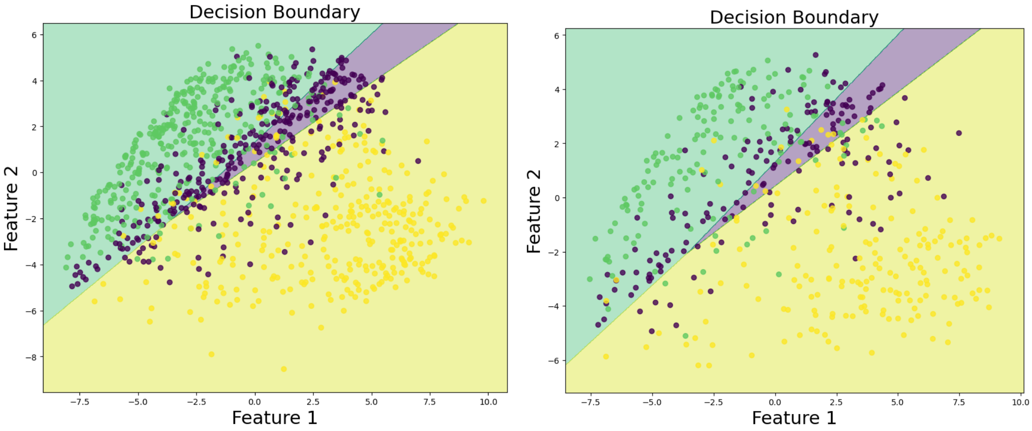

Figure 13: The data was reduced to 2 dimensions using PCA. (a) Inference on the training data to check if there is any overfitting. (b) Plotting the decision boundary for the test dataset.

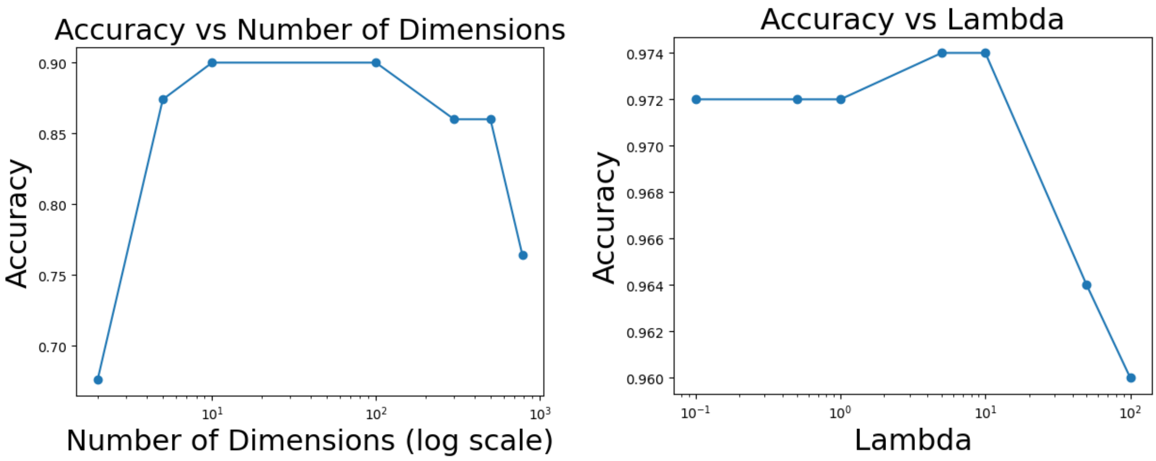

Figure 14: Hyper-parameter Tuning for a) Dimensions, b) Regularization Constant in Fashion MNIST