Clustering Assignment: Unsupervised Image Clustering

Course Name: Machine Learning

Assignment Title: Image Clustering with Textual Constraints

Submission Deadline: 22/3/2025

Total Marks: 100

Objective

This assignment aims to apply clustering algorithms on a standard image dataset and evaluate the performance with a textual constraint: cluster the images based on visual features and textual information hidden in the data.

Dataset

Use the CIFAR-10 dataset, which contains 60,000 color images in 10 classes, with 6,000 images per class. The dataset is publicly available via TensorFlow/Keras or PyTorch libraries.

Instructions

Pre-processing

1. Normalize the images.
2. Resize the images to 32x32 (if required).
3. Extract both visual features using any pre-trained CNN (ResNet-18 or VGG-16 or others) and semantic features using image captions (using CLIP or BLIP or other models).

Clustering with Visual Features (20 Marks)

1. Use K-Means and GMM Clustering on the image embeddings.
2. Assign the label of each data point in a cluster to be the class that has the maximum number of instances in that cluster.
3. Evaluate clustering performance using the Cohen Kappa Score.

Clustering with Textual Features (20 Marks)

1. Use the captions to extract text embeddings using Sentence-BERT or CLIP text encoder.
2. Perform clustering on the text embeddings.
3. Assign the label of each data point in a cluster to be the class that has the maximum number of instances in that cluster.

4. Evaluate clustering performance using the Cohen Kappa Score.

Fusion of Features (20 Marks)

1. Combine both visual and semantic embeddings (concatenation or weighted average).
2. Perform clustering again and assign labels as in previous steps.
3. Evaluate clustering performance using the Cohen Kappa Score.

Bonus (10 Marks)

Visualize the clusters using t-SNE.

Submission Guidelines

Upload a Jupyter Notebook file (.ipynb) with proper markdown and comments. Include all visualizations and performance metrics. The final report should contain: **(30 Marks)**

1. Dataset Description
2. Pre-processing Details
3. Clustering Algorithms Applied
4. Results and Discussion
5. Conclusion

Here's an example of how the **Cohen Kappa Score** evaluation should be done for the clustering assignment:

Example Evaluation Process

1. **Assign Cluster Labels:**
    - After performing clustering, each data point will belong to a cluster.
    - For each cluster, determine the **majority class label** (the most common ground truth label in that cluster).
    - Assign this majority class label to all points in that cluster.
2. Example: Suppose you have:

Cluster	Ground Truth Labels in Cluster	Majority Label
C1	[Cat, Cat, Dog, Cat]	Cat
C2	[Dog, Dog, Cat, Dog]	Dog

C3	[Bird, Bird, Bird, Cat]	Bird

3. **Predicted Labels:** Assign the **majority label** to all points in the cluster:
   - C1 → Cat
   - C2 → Dog
   - C3 → Bird
4. Predicted Labels: `['Cat', 'Cat', 'Cat', 'Cat', 'Dog', 'Dog', 'Dog', 'Dog', 'Bird', 'Bird', 'Bird', 'Bird']`
   Ground Truth: `['Cat', 'Cat', 'Dog', 'Cat', 'Dog', 'Dog', 'Cat', 'Dog', 'Bird', 'Bird', 'Bird', 'Cat']`

Cohen Kappa Score Calculation: Use Scikit-Learn to calculate the Cohen Kappa Score:

```
from sklearn.metrics import cohen_kappa_score

ground_truth = ['Cat', 'Cat', 'Dog', 'Cat', 'Dog', 'Dog', 'Cat', 'Dog', 'Bird', 'Bird', 'Bird', 'Cat']
predicted = ['Cat', 'Cat', 'Cat', 'Cat', 'Dog', 'Dog', 'Dog', 'Dog', 'Bird', 'Bird', 'Bird', 'Bird']

kappa_score = cohen_kappa_score(ground_truth, predicted)
print(f"Cohen Kappa Score: {kappa_score:.4f}")
```

5. ## Interpretation

- **Cohen Kappa Score = 1.0** → Perfect Agreement
- **Cohen Kappa Score = 0.0** → Random Agreement
- **Cohen Kappa Score < 0** → Worse than Random

For any queries, please send an email to 'shubhadipnag5555@gmail.com'
Demo Notebook:
https://colab.research.google.com/drive/19daFwpCgCu_i17k0X8HGhrh1x-_R-ymX?usp=sharing
Output Format (has been mentioned in the notebook):

```
Cohen Kappa Score (K-Means - Visual): 0.26010244735344346
Cohen Kappa Score (GMM - Visual): 0.26010244735344346
```