ClusterGAN: Latent Space Clustering in Generative Adversarial Networks

Final Project Report – Data Mining and Learning Systems

1. Introduction

Generative Adversarial Networks (GANs) are well-known for their ability to model complex data distributions and generate high-fidelity synthetic data. However, standard GAN architectures do not inherently support clustering in the latent space, which is a crucial requirement for many unsupervised learning tasks. In this project, we studied ClusterGAN, a novel architecture that enables clustering directly within the GAN's latent space while preserving sample quality and interpolation properties.

2. Motivation and Problem Statement

Standard GANs rely on a continuous prior (e.g., Gaussian or Uniform distribution) in the latent space, which spreads data points smoothly, making clustering ineffective. Moreover, traditional latent variables do not encode categorical or class-related information explicitly. Our goal was to explore whether GANs can be modified to:
- Support clustering in the latent space,
- Maintain interpolation capabilities across different classes,
- Retain high generation quality.

3. Proposed Approach (ClusterGAN)

ClusterGAN addresses the above limitations using three key ideas:
- Latent Prior Design: A discrete-continuous mixture prior is used: a one-hot encoded vector for discrete class encoding and a Gaussian noise vector for continuous variation.
- Inverse Mapping Network (Encoder): An encoder is jointly trained with the generator to map data points back to the latent space, aiding both reconstruction and clustering.
- Clustering-Specific Loss: The GAN is trained with additional regularization terms that enforce closeness between original and reconstructed latent vectors, as well as classification alignment for the discrete latent part.

4. Project Flow

1. Literature Review: Studied foundational works on GANs, clustering techniques, InfoGAN, and Autoencoders.
2. Understanding ClusterGAN: We deeply analyzed the architecture, latent design, and training methodology.

3. Reproducing Results: Implemented ClusterGAN using PyTorch and ran experiments on multiple datasets.
4. Evaluation: Compared ClusterGAN to baseline methods (e.g., InfoGAN, KMeans, NMF) using clustering metrics (ACC, NMI, ARI) and FID for sample quality.

5. Results

We evaluated ClusterGAN on several datasets including MNIST, Fashion-MNIST, Pendigits, and 10x73k (gene expression). The key observations:

Dataset	ACC (ClusterGAN)	NMI(Normalized Mutual Information)	ARI(Adjusted Rand Index)
MNIST	0.95	0.89	0.89
Fashion-10	0.63	0.64	0.50
10x 73k	0.81	0.73	0.67
Pendigits	0.77	0.73	0.65

6. Conclusions

ClusterGAN demonstrates that clustering and high-quality generation are not mutually exclusive in GANs. Through a carefully crafted latent space, combined with an inverse encoder and loss regularization, the model learns to cluster data unsupervised with excellent accuracy and stability.

7. Future Work

- Automatic estimation of cluster number.
- Learning data-driven latent priors.
- Better interpretability of learned clusters.
- Application to sparse and high-dimensional domains (e.g., genomics, NLP).