Generative Deep-Neural-Network Mixture Modeling with Semi-Supervised MinMax+EM Learning

Nilay Pande
Computer Science & Engineering

Suyash Awate
Computer Science & Engineering

Introduction: We tackle the problem of non-linear generative mixture modeling which has various applications in the field of image analysis such as clustering, interpolation or data generation. DNN based nonlinear generative mixture modeling typically rely on unsupervised learning that employs hard clustering schemes, or variational learning with loose/approximate bounds, or under-regularized modeling. We propose a novel statistical framework for a DNN mixture model using a single GAN, i.e., one generator, one encoder, and one discriminator.

Contributions: We design a novel data-likelihood term relying on a well-regularized/constrained Gaussian mixture model in the latent space along with a prior term on the DNN weights. We propose a novel learning formulation by combining min-max learning with EM-based learning, termed MinMax+EM, leveraging a variational lower bound that analytically guarantees tightness to the log-likelihood of the data. We extend our formulation to the semi-supervised setting and show superior results over the SOTA

Notations:

Image Data: $\{X_n\}_{n=1}^{N}$ with pdf $P(X)$

Latent space modeling: Multivariate gaussian pdf $P(Y)$ with k fixed means $\{\mu_k \in \mathbb{R}^L\}_{k=1}^{K}$, each with identity covariance and trainable scaling factors $\omega := \{\omega_k\}_{k=1}^{K}$

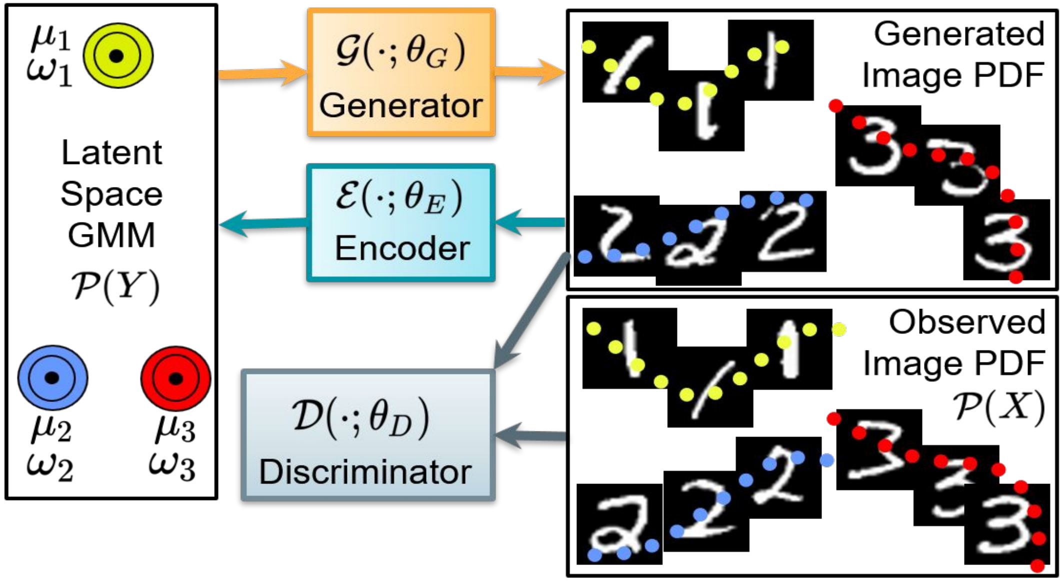
Hidden categorical random variable: Z $\in$[1, K], indicating the mixture component to which image X belongs

Generator, Encoder & Discriminator mappings: $\mathcal{G}(\cdot;\theta_G)$, $\mathcal{E}(\cdot;\theta_E)$ & $\mathcal{D}(\cdot;\theta_D)$ respectively

Posterior memberships at time t:

$\gamma_{nk}^t = P(Z = k | X, \theta_E^t, \omega^t)$

Architecture:



Learning objective terms:

Data log likelihood term

$$E_{P(X)} \log \left[\sum_{k=1}^{K} \omega_k \mathcal{N}(\mathcal{E}(X;\theta_E);\mu_k,\mathbf{I}) \right]$$

Consistency prior term on Generator+Encoder to ensure they are inverses of each other

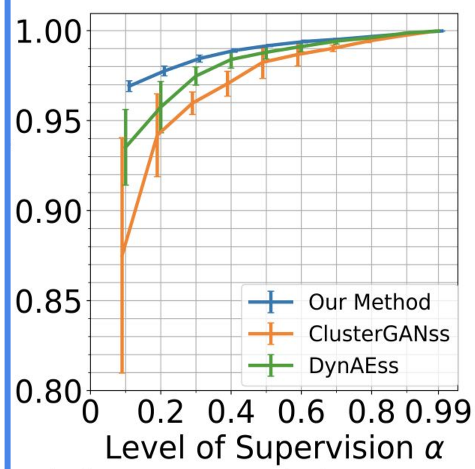
$$E_{P(Y)}[-\|Y-\mathcal{E}(\mathcal{G}(Y;\theta_G);\theta_E)\|_2^2]$$
$$= \sum_{k=1}^{K} \omega_k E_{Y_k\sim\mathcal{N}(\mu_k,\mathbf{I})}[-\|Y_k-\mathcal{E}(\mathcal{G}(Y_k;\theta_G);\theta_E)\|_2^2]$$

Gan loss term (Min-Max objective in GANs)

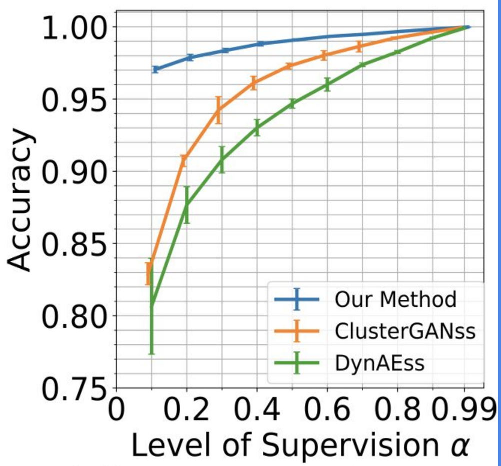
$$E_{P(X)}[-\log \mathcal{D}(X;\theta_D)] + E_{P(Y)}[\log \mathcal{D}(\mathcal{G}(Y;\theta_G);\theta_D)]$$
$$= E_{P(X)}[-\log \mathcal{D}(X;\theta_D)]$$
$$+ \sum_{k=1}^{K} \omega_k E_{Y_k\sim\mathcal{N}(\mu_k,\mathbf{I})}[\log \mathcal{D}(\mathcal{G}(Y_k;\theta_G);\theta_D)],$$

Results

We evaluate our method on 3 datasets- CIFAR10, CelebA and MNIST on varying levels of supervision and compare clustering accuracy against semi-supervised versions of ClusterGAN and DynAE



(a1) Accuracy: 5 clusters



(a2) Accuracy: 7 clusters

MinMax+EM Learning Algorithm:

To simplify the data log-likelihood term, we use EM to design an optimal lower bound on the data log-likelihood that touches the log-likelihood function at the current parameter estimate in the E step, and update the parameters to improve the value of the bound in the M step

1) **Optimal lower bound:**

$$Q(\theta_E,\omega;\theta_E^t,\omega^t)$$
$$:= E_{P(X)}E_{P(Z|X,\theta_E^t,\omega^t)}[\log P(X,Z|\theta_E,\omega)]$$
$$= E_{P(X)}\left[\sum_{k=1}^{K} P(Z=k|X,\theta_E^t,\omega^t)\log P(X|Z=k,\theta_E,\omega)\right]$$
$$+ E_{P(X)}\left[\sum_{k=1}^{K} P(Z=k|X,\theta_E^t,\omega^t)\log \omega_k\right],$$

2) **Final MinMax+EM Learning objective**

Combining optimal EM lower bound with consistency prior and MinMax GAN loss term, we get the final MinMax objective as-

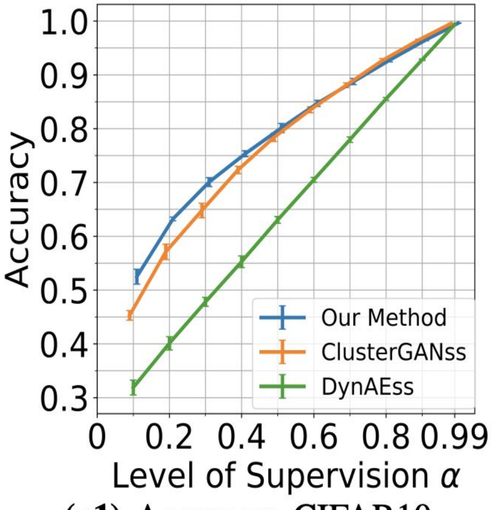
$$\min_{\theta_D} \max_{\omega,\theta_G,\theta_E} \sum_{n=1}^{N}\sum_{k=1}^{K} \gamma_{nk}^t \left(\log\omega_k + \log\mathcal{N}(\mathcal{E}(x_n;\theta_E);\mu_k,\mathbf{I})\right)$$
$$-\lambda_1 \sum_{k=1}^{K}\omega_k \sum_{s=1}^{S} \|y_k^s - \mathcal{E}(\mathcal{G}(y_k^s;\theta_G);\theta_E)\|_2^2$$
$$-\lambda_2 \sum_{n=1}^{N} \log \mathcal{D}(x_n;\theta_D)$$
$$+\lambda_2 \sum_{k=1}^{K}\omega_k \sum_{s=1}^{S} \log \mathcal{D}(\mathcal{G}(y_k^s;\theta_G);\theta_D).$$

3) **Extension to semi-supervised case**

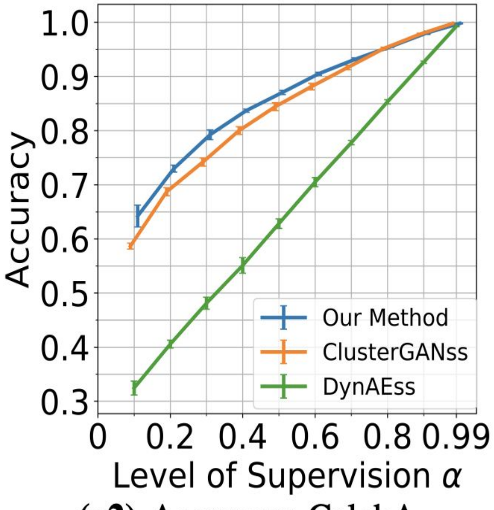
In this case, since we have labels for some of the images, we consider memberships for those images to be crisp and thus, we additionally have a supervised loss term for the encoder, mapping those images to their respective cluster centers in latent space.

Results Continued

CIFAR10 & CelebA (5 classes):



(a1) Accuracy: CIFAR10



(a2) Accuracy: CelebA

Conclusion

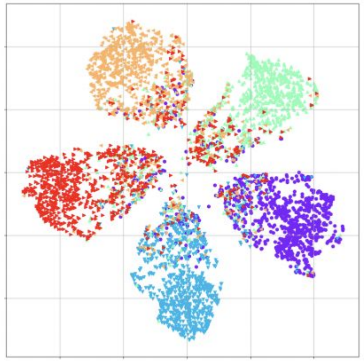
Our GMM-based data-likelihood maximizing formulation leads to statistically significantly better performance than ClusterGANss and DynAEss, especially at smaller levels of supervision α, indicating improved robustness to noise, for all the datasets.

The t-SNE visualizations of the latent-space PDFs clearly indicate that the proposed method produces image encodings for each mixture component with far smaller overlap across components.

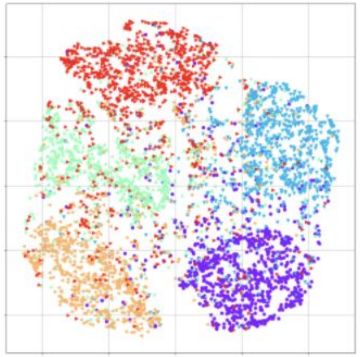
Results on three real-world image datasets demonstrate the benefits of our compact modeling and learning formulation over the state of the art for nonlinear generative image (mixture) modeling and image clustering.

t-SNE visualizations for latent space PDFs at 0.5 supervision

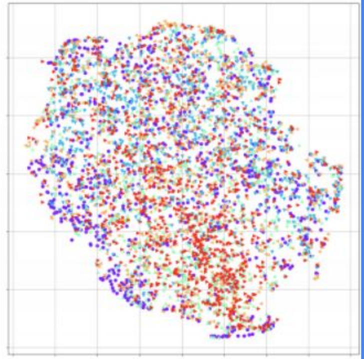
CIFAR10 (5 classes):



(a1) Ours: CIFAR10

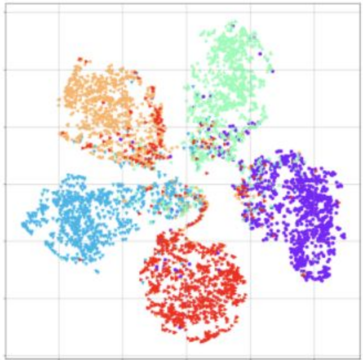


(b1) ClusterGANss: CIFAR10

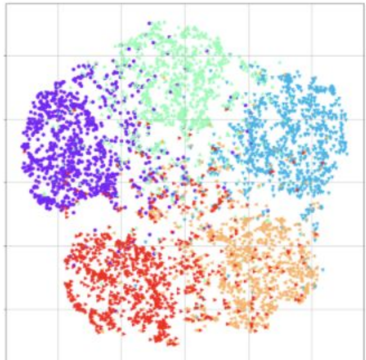


(c1) DynAEss: CIFAR10

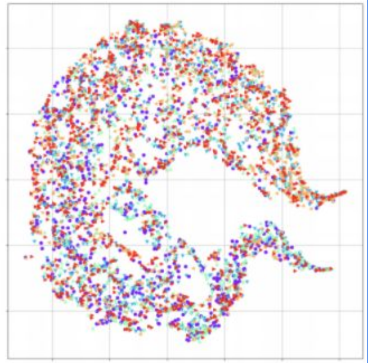
CelebA (5 classes):



(a2) Ours: CelebA



(b2) ClusterGANss: CelebA



(c2) DynAEss: CelebA

References

1) S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, “ClusterGAN: Latent space clustering in generative adversarial networks,” in AAAI Conf. Artific. Intell., 2019, pp. 4610–7.
2) K. Ghasedi, X. Wang, C. Deng, and H. Huang, “Balanced self-paced learning for generative adversarial clustering network,” in IEEE Comp. Vis. Patter. Recog., 2019, pp. 4386–4395.
3) L. Yang, N. Cheung, J. Li, and J. Fang, “Deep clustering by Gaussian mixture variational autoencoders with graph embedding,” in Int. Conf. Comp. Vis., 2019, pp. 6439–48.
4) N. Mrabah, N. Khan, and R. Ksantini, “Deep clustering with a dynamic autoencoder,” arxiv.org/abs/1901.07752, 2019