Deep DPM: Deep Clustering With an Unknown Number of Clusters

2023. 01. 18

Presenter : Taewoong Kang
twk@deepnoid.com

DEEP NOID

Abstract

- Effective Deep-clustering method
    - (When K is unknown, model become computationaly expensive)
- Does not require knowing the value of K
- Using split/merging framework, a dynamic architecture, novel loss

Figure 3. Examples of ImageNet images clustered together by DeepDPM. Each panel stands for a different cluster.

Architecture

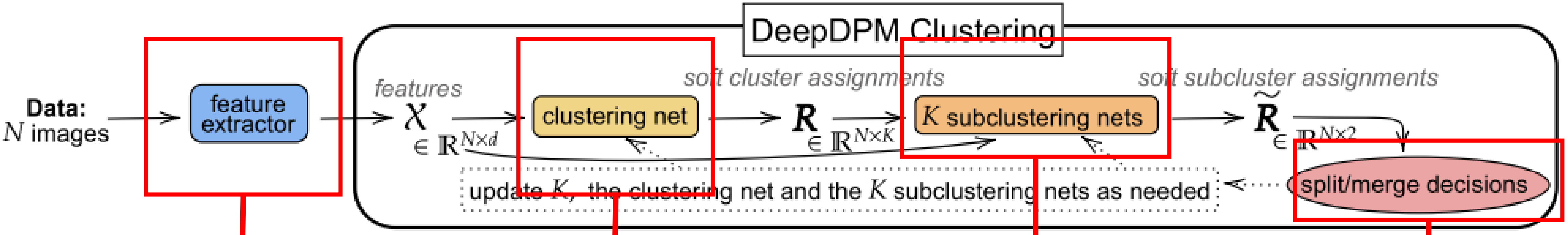

Figure 2. DeepDPM's pipeline: given features $\mathcal{X}$, the clustering net outputs cluster assignments, $\boldsymbol{R}$, while the subclustering nets generate subcluster assignments, $\widetilde{\boldsymbol{R}}$. Upon the acceptance of split/merge proposals, all those nets are updated during the learning.

Embedding
(use MOCO)

Cluster A, B,.., K

Cluster A1, A2
…
Cluster K1, K2

Change K

DEEP NOID

Notations

$\mathcal{X} = (\boldsymbol{x}_i)_{i=1}^N$ denote N data points in $\mathbb{R}^d$

z_i is the point-to-cluster assignment Cluster label $(\boldsymbol{x}_i)_{i:z_i=k}$.

$K \triangleq |\{k : k \in \boldsymbol{z}\}|$

$\boldsymbol{z} = (z_i)_{i=1}^N$.

DPGMM (the Dirichlet Process Gaussian Mixture Model)

- BNP (Bayesian nonparametric) extension of GMM

$$p(\boldsymbol{x}|(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)_{k=1}^{\infty}) = \sum_{k=1}^{\infty} \pi_k \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

Notations

- Gaussian pdf (probability density function) $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
- mean $\boldsymbol{\mu}_k \in \mathbb{R}^d$ and a d-by-d covariance matrix $\boldsymbol{\Sigma}_k$
- $\boldsymbol{x} \in \mathbb{R}^d$, $\pi_k > 0\ \forall k$, and $\sum_{k=1}^{\infty} \pi_k = 1$
- $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
- $\boldsymbol{\theta} = (\boldsymbol{\theta}_k)_{k=1}^{\infty}$ are iid, draws from their prior, typically a NIW distribution
- $\boldsymbol{\pi} = (\pi_k)_{k=1}^{\infty}$ are drawn using the GEM (the Griffiths-Engen-McCloskey stick-breaking process)
- $\alpha > 0$ the concentration parameter

Metropolis-Hastings framework

- Hasting Ratio

$$H_s = \frac{\alpha\Gamma(N_{k,1})f_{\boldsymbol{x}}(\mathcal{X}_{k,1};\lambda)\Gamma(N_{k,2})f_{\boldsymbol{x}}(\mathcal{X}_{k,2};\lambda)}{\Gamma(N_k)f_{\boldsymbol{x}}(\mathcal{X}_k;\lambda)}$$

Γ is the Gamma function

$N_k = |\mathcal{X}_k|$

$f_{\boldsymbol{x}}(\cdot;\lambda)$ Is the marginal likelihood where λ represents the NIW hyperparameters

NIW

- A conjugate prior to the multivariate normal distribution with an unknown mean and unknown covariance matrix -〉 posterior probability will be in the same distribution
- Algebraically convenient to inference

- Pdf of Inverse-Wishart(IW)

$$\mathcal{W}^{-1}(\mathbf{\Sigma}_k;\nu,\mathbf{\Psi}) = \frac{|\nu\mathbf{\Psi}|^{\frac{\nu}{2}}}{2^{\frac{\nu d}{2}}\Gamma_d(\frac{\nu}{2})}|\mathbf{\Sigma}_k|^{-\frac{\nu+d+1}{2}}e^{-\frac{1}{2}\operatorname{tr}(\nu\mathbf{\Psi}\mathbf{\Sigma}_k^{-1})}$$

$\nu > d-1$, $\mathbf{\Psi} \in \mathbb{R}^{d\times d}$ is SPD, and Γ_d is the (d-dimensional) multivariate gamma function.

The positive real number ν and the SPD matrix $\mathbf{\Psi}$ are called the hyperparameters

- Symmetric and Positive Definite(SPD)

NIW

- Pdf of Normal-Inverse-Wishart(NIW)

$$p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k; \kappa, \boldsymbol{m}, \nu, \boldsymbol{\Psi}) = \mathrm{NIW}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k; \kappa, \boldsymbol{m}, \nu, \boldsymbol{\Psi}) \triangleq \overbrace{\mathcal{N}(\boldsymbol{\mu}_k; \boldsymbol{m}, \tfrac{1}{\kappa}\boldsymbol{\Sigma}_k)}^{p(\boldsymbol{\mu}_k | \boldsymbol{\Sigma}_k; \kappa, \boldsymbol{m})} \overbrace{\mathcal{W}^{-1}(\boldsymbol{\Sigma}_k; \nu, \boldsymbol{\Psi})}^{p(\boldsymbol{\Sigma}_k; \nu, \boldsymbol{\Psi})}$$

$\boldsymbol{m} \in \mathbb{R}^d$ and $\kappa > 0$ (while ν and $\boldsymbol{\Psi}$ are as before) and $\mathcal{N}(\boldsymbol{\mu}_k; \boldsymbol{m}, \frac{1}{\kappa}\boldsymbol{\Sigma}_k)$ is a d-dimensional Gaussian pdf

$$(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \sim \mathrm{NIW}(\boldsymbol{m}, \kappa, \boldsymbol{\Psi}, \nu) \qquad \lambda \triangleq (\boldsymbol{m}, \kappa, \boldsymbol{\Psi}, \nu)$$

- Posterior hyperparameters

$$p(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k | \mathcal{X}_k) = \mathrm{NIW}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k; \kappa^*, \boldsymbol{m}_k^*, \nu^*, \boldsymbol{\Psi}_k^*)$$

$$\kappa_k^* = \kappa + N_k$$

$$\boldsymbol{m}_k^* = \frac{1}{\kappa_k^*}\left[\kappa \boldsymbol{m} + \sum_{i:z_i=k} \boldsymbol{x}_i\right]$$

$$\nu_k^* = \nu + N_k$$

$$\boldsymbol{\Psi}_k^* = \frac{1}{\nu^*}\left[\nu\boldsymbol{\Psi} + \kappa \boldsymbol{m}\boldsymbol{m}^T + \left(\sum_{i:z_i=k} \boldsymbol{x}_i \boldsymbol{x}_i^T\right) - \kappa_k^* \boldsymbol{m}_k^* (\boldsymbol{m}_k^*)^T\right]$$

DEEP NOID

Marginal Likelihood Function

- When marginalizing over the parameters of the Gaussian, one obtains the marginal data likelihood

$$f_{\boldsymbol{x}}((\boldsymbol{x}_i)_{i=1}^N;\lambda) = f_{\boldsymbol{x}}((\boldsymbol{x}_i)_{i=1}^N;\boldsymbol{m},\kappa,\boldsymbol{\Psi},\nu) = \int p((\boldsymbol{x}_i)_{i=1}^N|\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)p(\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k;\lambda)d(\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$$

$$= \frac{1}{\pi^{\frac{Nd}{2}}} \frac{\Gamma_d(\nu^*/2)}{\Gamma_d(\nu/2)} \frac{|\nu\boldsymbol{\Psi}|^{\nu/2}}{|\nu^*\boldsymbol{\Psi}_k^*|^{\nu^*/2}} \left(\frac{\kappa}{\kappa^*}\right)^{d/2}$$

where Γ_d is the d-dimensional Gamma function.

DeepDPM

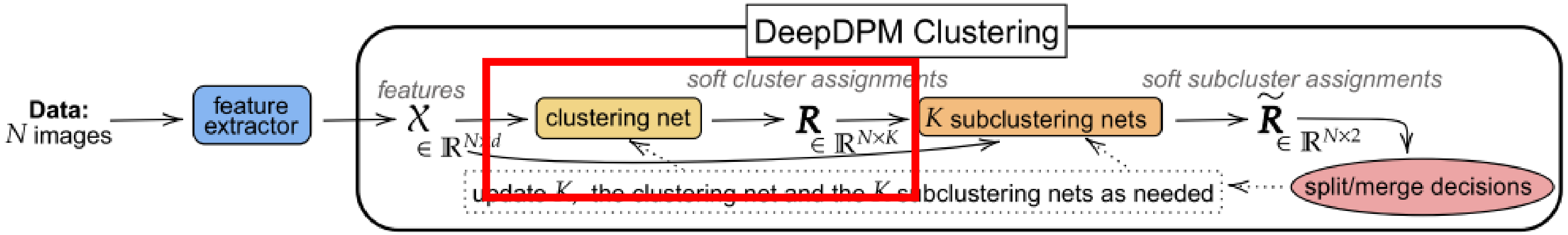

Figure 2. DeepDPM's pipeline: given features $\mathcal{X}$, the clustering net outputs cluster assignments, $\boldsymbol{R}$, while the subclustering nets generate subcluster assignments, $\widetilde{\boldsymbol{R}}$. Upon the acceptance of split/merge proposals, all those nets are updated during the learning.

- Clustering net
  - Given the current value K, the data is passed to the clustering net.
  - Generates K soft cluster assignments

$$f_{\text{cl}}(\mathcal{X}) = \boldsymbol{R} = (\boldsymbol{r}_i)_{i=1}^{N} \qquad \boldsymbol{r}_i = (r_{i,k})_{k=1}^{K}$$

  - We compute the hard assignments

$$\boldsymbol{z} = (z_i)_{i=1}^{N} \text{ by } z_i = \arg\max_k r_{i,k}.$$

$$p(z_k = 1|\mathbf{x}_n) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\mu_j, \Sigma_j)} = \gamma(z_{nk})$$

DeepDPM

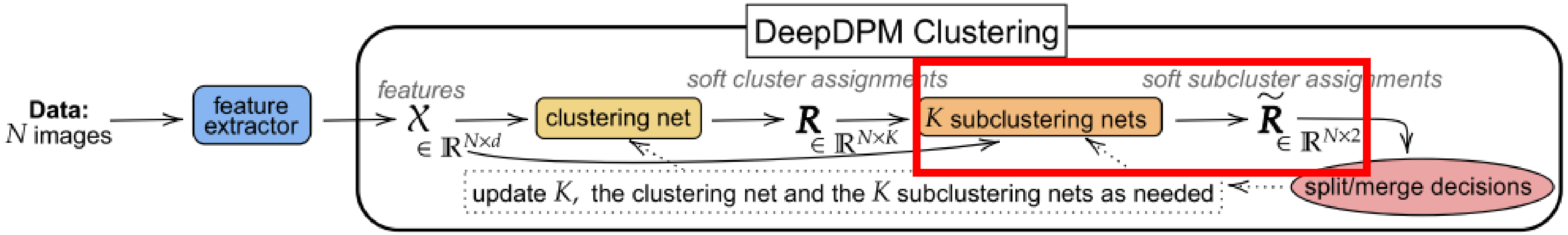

Figure 2. DeepDPM's pipeline: given features $\mathcal{X}$, the clustering net outputs cluster assignments, $\boldsymbol{R}$, while the subclustering nets generate subcluster assignments, $\widetilde{\boldsymbol{R}}$. Upon the acceptance of split/merge proposals, all those nets are updated during the learning.

- Subclustering net
  - Each subclustering net is fed with the data (hard-) assigned to its respective cluster
  - Generates soft subcluster assignments

$$f_{\text{sub}}^{k}(\mathcal{X}_k) = \widetilde{\boldsymbol{R}}_k = (\widetilde{\boldsymbol{r}}_i)_{i:z_i=k} \qquad \widetilde{\boldsymbol{r}}_i = (\widetilde{r}_{i,j})_{j=1}^{2}$$

Splits and Merges

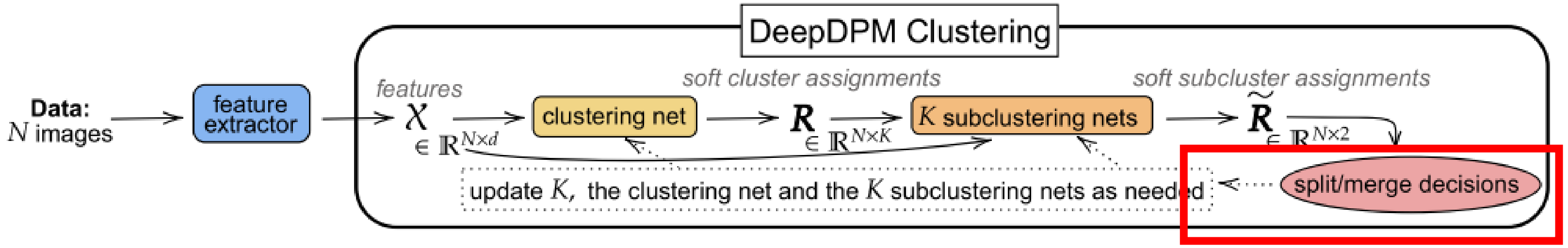

Figure 2. DeepDPM's pipeline: given features $\mathcal{X}$, the clustering net outputs cluster assignments, $\boldsymbol{R}$, while the subclustering nets generate subcluster assignments, $\widetilde{\boldsymbol{R}}$. Upon the acceptance of split/merge proposals, all those nets are updated during the learning.

$$\boldsymbol{\mu}_{k_1} \leftarrow \widetilde{\boldsymbol{\mu}}_{k,1}\,, \quad \boldsymbol{\Sigma}_{k_1} \leftarrow \widetilde{\boldsymbol{\Sigma}}_{k,1}\,, \quad \pi_{k_1} \leftarrow \pi_k \times \widetilde{\boldsymbol{\pi}}_{k,1}$$
$$\boldsymbol{\mu}_{k_2} \leftarrow \widetilde{\boldsymbol{\mu}}_{k,2}\,, \quad \boldsymbol{\Sigma}_{k_2} \leftarrow \widetilde{\boldsymbol{\Sigma}}_{k,2}\,, \quad \pi_{k_2} \leftarrow \pi_k \times \widetilde{\boldsymbol{\pi}}_{k,2}$$

- Changing K via Splits and Merges
  - Splits
    - A split proposal is accepted stochastically with probability $\min(1, H_\mathrm{s})$
  - Merges
    - Must ensure we never mistakenly merging three clusters together
    - Consider the merges of each cluster with only its 3 nearest neighbors
    - Merge proposal is accepted/rejected using a Hasting ratio $H_\mathrm{m} = 1/H_\mathrm{s}$
    - The parameter and the weight of the new-cluster are initialized using the weighted MAP estimation

Loss function

- Motivated by EM (Expectation Maximization) in the Bayesian GMM

- E step
  - Responsibility (책임값)
  - Each E step Is followed by a standard M step in the Bayesian GMM (except soft assignment used in MAP)

$$r_{i,k}^{\mathrm{E}} = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})} \quad k \in \{1, \ldots, K\}$$

$(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)_{k=1}^{K}$ from the previous epoch

$\sum_{k=1}^{K} r_{i,k}^{\mathrm{E}} = 1$

$$\mathcal{L}_{\mathrm{cl}} = \sum_{i=1}^{N} \mathrm{KL}(\boldsymbol{r}_i \| \boldsymbol{r}_i^{\mathrm{E}})$$

$$\mathcal{L}_{\mathrm{sub}} = \sum_{k=1}^{K} \sum_{i=1}^{N_k} \sum_{j=1}^{2} \tilde{r}_{i,j} \| \boldsymbol{x}_i - \tilde{\boldsymbol{\mu}}_{k,j} \|_{\ell_2}^2$$

DEEP NOID

Loss function

- Bayesian M step
  - Use weighted versions of the MAP estimates of $(\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)_{k=1}^{K}$
  - Instead of $r_{i,k}^{\mathrm{E}}$ we use $r_{i,k}$ as weights

DEEP NOID

Weighted MAP Estimates

- In the unweighted case, we use
$$\boldsymbol{\Sigma}_k = \frac{\nu^* \boldsymbol{\Psi}_k^*}{\nu^* - d + 1}$$
$$\boldsymbol{\mu}_k = \boldsymbol{m}_k^* .$$
\`\`\`Eq1

- In weighted MAP, we still use Eq1, but instead of the posterior hyperparameters we use their weighted versions

$$\kappa_k^* = \kappa + N_k$$
$$\boldsymbol{m}_k^* = \frac{1}{\kappa_k^*}\left[\kappa \boldsymbol{m} + \sum_{i:z_i=k} \boldsymbol{x}_i\right]$$
$$\nu_k^* = \nu + N_k$$
$$\boldsymbol{\Psi}_k^* = \frac{1}{\nu^*}\left[\nu\boldsymbol{\Psi} + \kappa \boldsymbol{m}\boldsymbol{m}^T + \left(\sum_{i:z_i=k} \boldsymbol{x}_i\boldsymbol{x}_i^T\right) - \kappa_k^* \boldsymbol{m}_k^* (\boldsymbol{m}_k^*)^T\right] .$$

$\Rightarrow$

$$\kappa_k^* = \kappa + \sum_{i=1}^N r_{i,k}$$
$$\boldsymbol{m}_k^* = \frac{1}{\kappa_k^*}\left[\kappa \boldsymbol{m} + \sum_{i=1}^N r_{i,k}\boldsymbol{x}_i\right]$$
$$\nu_k^* = \nu + \sum_{i=1}^N r_{i,k}$$
$$\boldsymbol{\Psi}_k^* = \frac{1}{\nu^*}\left[\nu\boldsymbol{\Psi} + \kappa \boldsymbol{m}\boldsymbol{m}^T + \left(\sum_{i=1}^N r_{i,k}\boldsymbol{x}_i\boldsymbol{x}_i^T\right) - \kappa_k^* \boldsymbol{m}_k^* (\boldsymbol{m}_k^*)^T\right]$$

DEEP NOID

Amortized EM Inference

- Our method still yields results that are usually better than the standard EM
- By the virtue of the smoothness of the function learned by the deep net, we improve the prediction for the points in no only the current batch but also other batches
- The smoothness serves as an inductive bias, such that points should have similar labels
- When using the GMM negative log likelihood instead of our loss, empirically that led to unstable optimization and/or poor results

	ACC		
	K_{init}=3	K_{init}=10	K_{init}=30
No splits/merges	.29±.01	.59±.03	.46±.01
No splits	.29±.01	.59±.02	.45±.03
No merges	.46±.00	.58±.01	.47±.01
2-means instead of f_{sub}	.61±.00	.59±.02	.56±.02
No priors in the M step	.58±.01	.57±.02	.58±.01
Isotropic loss instead of $\mathcal{L}_{\text{cl}}$	.58±.00	.58±.00	.58±.02
DeepDPM (full method)	**.62±.03**	**.61±.00**	**.62±.01**

Table 6. DeepDPM's performance under different ablations.

- On Fashion-MNIST

DEEP NOID

A Weak Prior : Letting the Data Speak for Itself

- The inferred K depends on $\mathcal{X}$, α, and the NIW hyperparameters
- Intentionally choose the prior to be very weak
- When doing posterior calculation, if $\alpha \ll N$, where N is the number of data points, then the importance of α diminishes

- For example, v is very high and Ψ is small -> favor small clusters, thus K is likely to be high
- v is very high and Ψ is large -> favor large clusters, so K will tend to be small
- Our α, ν and κ are all much smaller than N in all the datasets

Results

Classical methods

- Nonparametric ones are less affected by the imbalance.

	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC
		MNIST [18]			USPS [35]			Fashion-MNIST [69]	
K-means^p	.90± .02	.84± .05	.85±.06	.86±.01	.79±.05	.80±.06	.67±.01	.50±.03	.60±.04
GMM^p	**.94±.00**	**.95±.00**	**.98±.00**	.86±.02	.79±.05	.81±.06	.66±.01	.49±.02	.58±.03
DBSCAN	.92±0	.86±0	.89±0	.72±0	.46±0	.57±0	.63±0	-.32±0	.39±0
DPM Sampler	.92±.01	.91±.04	.93±.05	.87±.01	.82±.02	.83±.03	.67±.01	.49± .02	.59±.03
moVB	.93±.00	.94±.00	.97±.00	.87±.02	**.86±.04**	**.90±.04**	.66±.02	.47±.03	.55±.03
DeepDPM (Ours)	**.94±.00**	**.95±.00**	**.98±.00**	**.88±.00**	**.86±.01**	.89±.2	**.68±.01**	**.51±.02**	**.62±.03**
		MNIST^{imb}			USPS^{imb}			Fashion-MNIST^{imb}	
K-means^p	.89± .03	.84± .06	.83±.06	.82±.02	.71±.05	.71±.05	.62±.01	.46±.02	.56±.03
GMM^p	.94±.02	.95±.03	.96±.04	.83±.01	.74±.05	.76±.05	.62±.01	.46±.02	.57±.03
DBSCAN	.93±0	.92±0	.94±0	.84±0	.79±0	.80±0	.62±0	.35±0	.46±0
DPM Sampler	.93±.01	.94±.02	.96±.02	.89±.02	.89±.06	.91±.04	**.66±.01**	**.50± .01**	**.61±.01**
moVB	.94±.00	.95±.00	.96±.00	.88±.01	.89±.02	.91±.02	.63±.01	.44±.02	.53±.02
DeepDPM (Ours)	**.95±.01**	**.97±.01**	**.98±.01**	**.90±.00**	**.92±.00**	**.94±.00**	.65±.00	**.50±.00**	**.61±.00**

Table 1. Comparing the mean results (±std. dev.) of DeepDPM with classical clustering methods. The results are the mean of 10 independent runs. Methods marked with ^{p} are parametric (require K). Datasets marked with ^{imb} are imbalanced ones.

Nonparametric method

- DPM's inferred K is the closest to GT K

Method	Inferred K		
	MNIST	USPS	Fashion-MNIST
DBSCAN	9.0±0.00	6.0±0.00	4.0±0.00
DPM Sampler	11.3±0.82	8.5±0.85	12.4±0.97
moVB	14±1.00	11.2±1.08	16.9±2.30
DeepDPM (Ours)	**10±0.00**	**9.2±0.42**	**10.2±0.79**

Table 2. Comparing the mean inferred value (±std. dev.) for K of 10 runs among nonparametric methods. GT $K = 10$.

DEEP NOID

Deep Nonparametric Methods

	MNIST [18]			STL-10 [15]			Reuters10k [43]		
Method	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC
AdapVAE† [74] *avg*	.86±1.02	.84±2.35	N/A	.75±0.53	**.71±0.81**	N/A	.45±1.79	.43±5.73	N/A
DCC† [52] best	.912	N/A	.96	N/A	N/A	N/A	.59	N/A	.60
DCC‡ [52] *avg*	.90±.02	.89±.07	.91±.07	.22±.00	.01±.00	.04±.00	.25±.00	.00±.00	.00±.00
DeepDPM (ours) *avg*	.90±.01	.91±.02	.93±.03	.78±.004	.70±.01	.84±.01	.61±.00	.64±.01	.83±.00
DeepDPM (ours) best	**.92**	**.93**	**.96**	**.79**	**.71**	**.85**	**.61**	**.64**	**.83**

Table 3. Comparing deep nonparametric methods. †: reported in the papers. ‡: obtained using their code. *avg*: mean (±std. dev.) of 5 runs.

The value of Deep Nonparametric Methods

Method	NMI	ARI	ACC
	ImageNet-50: Balanced		
DBSCAN	.52±.00	.09±.00	.24±.00
moVB	.70±.01	.38±.01	.55±.02
DPM Sampler	.72±.00	.43±.01	.57±.01
DeepDPM (ours)	.75±.00	.49±.01	.64±.00
DeepDPM (ours)∗	**.77±.00**	**.54±.01**	**.66±.01**
	ImageNet-50: Imbalanced		
DBSCAN	.33±.00	.04±.00	.24±.00
moVB	.68±.01	.44±.03	.52±.03
DPM Sampler	.70±.00	.40±.01	.51±.00
DeepDPM (ours)	.74±.01	.48±.02	.58±.01
DeepDPM (ours)∗	**.75±.00**	**.51±.01**	**.60±.01**

Table 4. Comparison of nonparametric methods on ImageNet-50 and its imbalanced version. ∗ marks results with AE alternation.

Method	Final/best K: balanced	Final/best K: imbalanced
K-means^p	40	20
DCN++^{p}	60	40
SCAN^p	70	40
DBSCAN	16	13
moVB	46.2±1.3	**46.4±1.1**
DPM Sampler	72.0±2.6	70.3±4.6
DeepDPM (ours)	**52.0±1.0**	43.67±1.2
DeepDPM (ours)∗	55.3±1.5	46.3±2.5

Table 5. Comparing the mean (±std. dev.) value for K found on ImageNet-50 of 3 runs. For the parametric methods (marked with ^{p}) we use the K value with the best silhouette score. ∗ marks results obtained with AE alternation.

Ablation Study and Robustness to the Initial K

	ACC		
	K_{init}=3	K_{init}=10	K_{init}=30
No splits/merges	.29±.01	.59±.03	.46±.01
No splits	.29±.01	.59±.02	.45±.03
No merges	.46±.00	.58±.01	.47±.01
2-means instead of f_{sub}	.61±.00	.59±.02	.56±.02
No priors in the M step	.58±.01	.57±.02	.58±.01
Isotropic loss instead of $\mathcal{L}_{\text{cl}}$	.58±.00	.58±.00	.58±.02
DeepDPM (full method)	**.62±.03**	**.61±.00**	**.62±.01**

Table 6. DeepDPM's performance under different ablations.

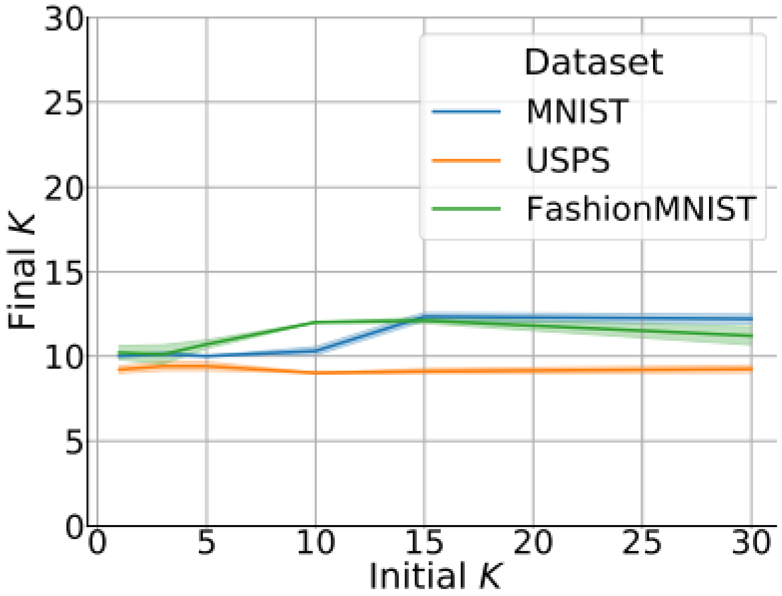

Figure 4. Robustness to the initial K. GT $K = 10$ in all datasets.

- On Fashion-MNIST

DEEP NOID

Limitations

- If Deep-DPM's input features are poor it would struggle to recover.
- K is known and dataset is balanced, parametric methods may be a slightly better choice.

Summary

- Outperforms deep and non-deep nonparametric methods and achieves SOTA results
- Demonstrated the added value the nonparametric approach brings to deep clustering

Clustering the Entire ImageNet Dataset. On ImageNet, we obtained the following results: ACC: 0.25, NMI: 0.65, ARI: 0.14. Our method was initialized with $K = 200$ and converged into 707 clusters (GT=1000). These are first results on ImageNet reported for deep nonparametric clustering. Figure 3 shows examples of images clustered together.

Q & A

감사합니다

Feature extraction

- Changing K via Splits and Merges
  - Splits
    - A split proposal is accepted stochastically with probability
  - Merges
    - Must ensure we never mistakenly