The DipEncoder: Enforcing Multimodality in Autoencoders

28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining
August 14-18, 2022, Washington, DC, USA

Collin Leiber[1], Lena G. M. Bauer[2], Michael Neumayr[1], Claudia Plant[2], Christian Böhm[2]

[1] Ludwig-Maximilians-Universität München, Munich, Germany

[2] Universität Wien, Vienna, Austria

Outline

- Motivation and Introduction
- Dip-test
- The DipEncoder
- Deep Clustering algorithm using the DipEncoder
- Experiments
- Conclusion

Motivation and introduction I

- Clustering large amounts of high-dimensional data causes problems for classical clustering methods

➔ Common solution:

Preprocess the data by using a dimensionality reduction technique

➔ Modern solution:

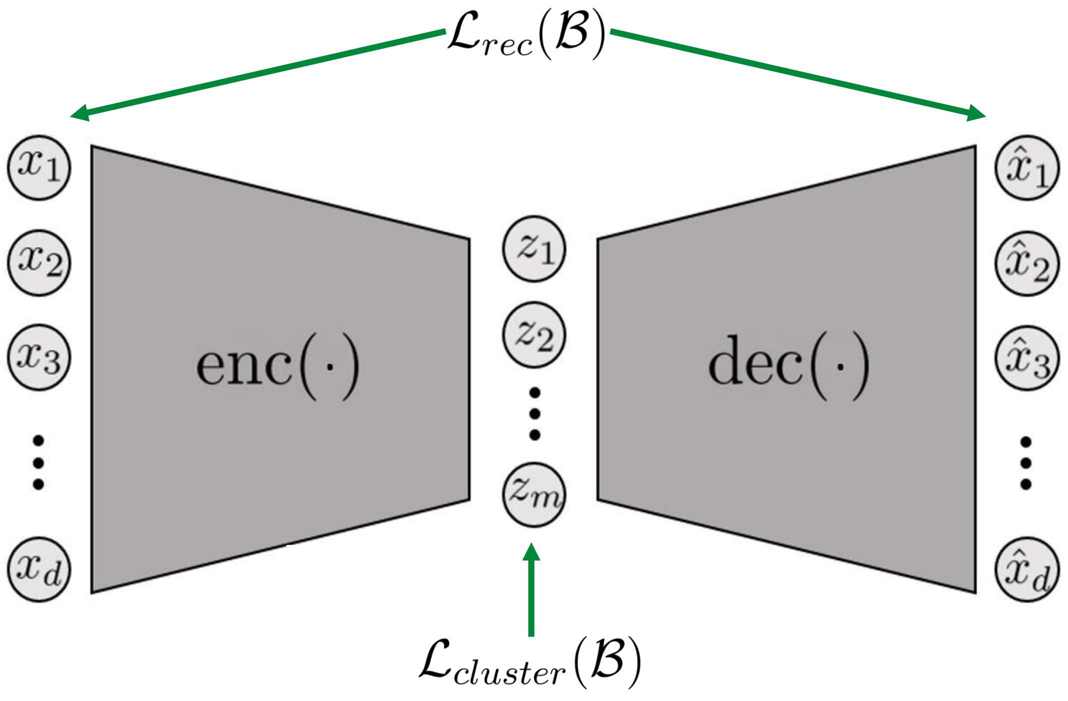
Perform dimensionality-reduction and clustering simultaneously using an autoencoder (**Deep Clustering**)

Motivation and introduction II

- Deep Clustering approaches usually optimize two losses:
  - $\mathcal{L}_{rec}(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} ||x - \hat{x}||_2^2$

    → Improves the embedding
  - $\mathcal{L}_{cluster}(\mathcal{B}) \rightarrow$ various losses have been presented

    → Improves the actual clustering



Motivation and introduction II

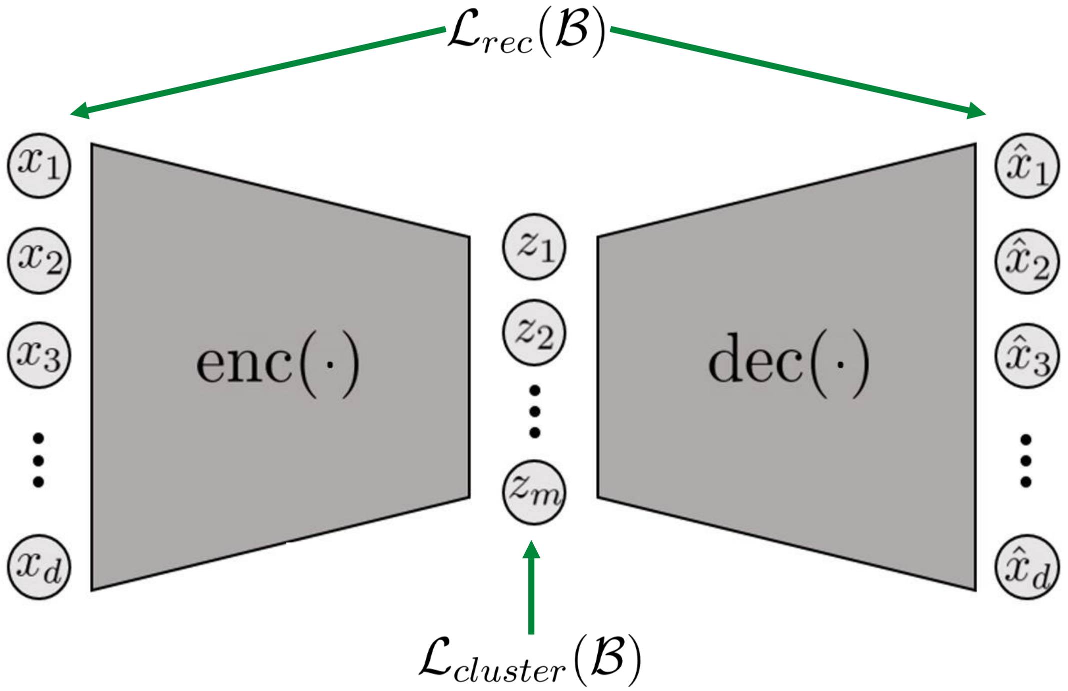
- Deep Clustering approaches usually optimize two losses:

  - $\mathcal{L}_{rec}(\mathcal{B}) = \frac{1}{|\mathcal{B}|}\sum_{x \in \mathcal{B}} ||x - \hat{x}||_2^2$

    → Improves the embedding

  - $\mathcal{L}_{cluster}(\mathcal{B}) \rightarrow$ various losses have been presented

    → Improves the actual clustering

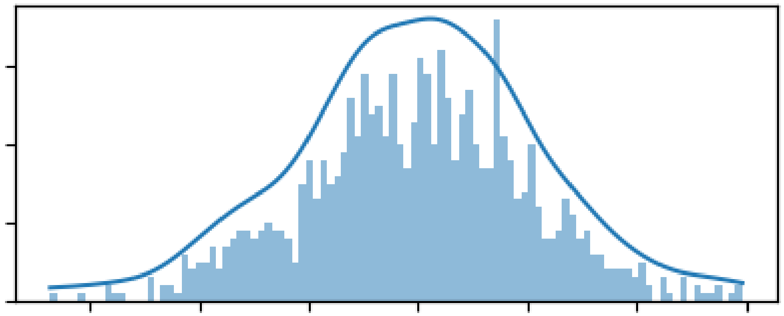


- Problem: Often assumptions about the structure of the clusters are necessary

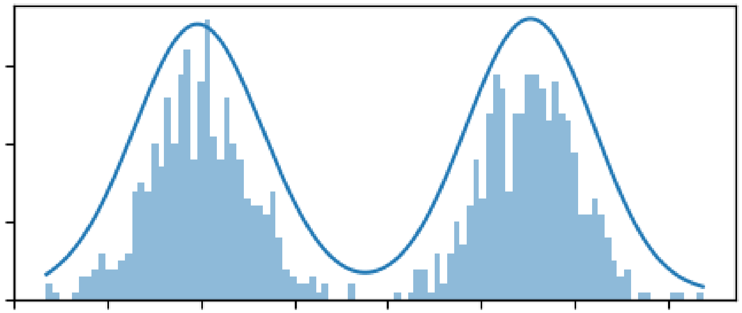
→ Use the dip-test to optimize embedding

The Dip-test of unimodality

- Measures modality in sorted one-dimensional samples
- $dip \in (0, 0.25]$
  - $dip \approx 0$ → unimodal
  - $0 \ll dip \leq 0.25$ → multimodal
- Makes no assumption about an underlying data distribution and is parameter-free



dip = 0.0096

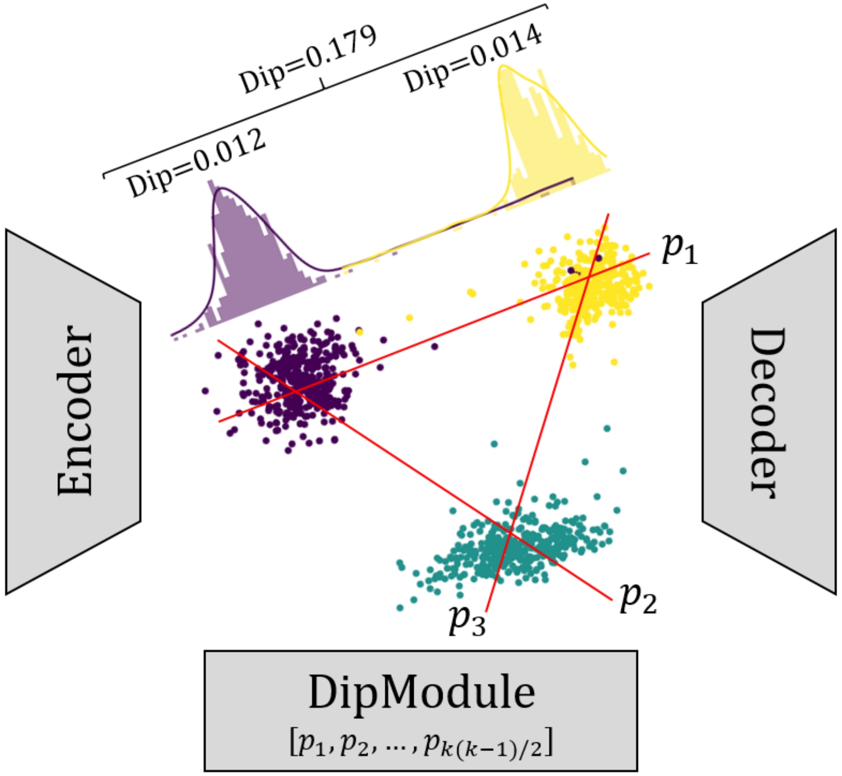


dip = 0.1103

DipEncoder I

- In multidimensional space, the Dip-test is usually performed with data projected onto a projection axis
- We create one projection axis $p_{a,b}$ for each combination of clusters
- Those axes are stored in a separate NN
  → The DipModule

- The update of the autoencoder should result in
  - **High** modality between **two clusters**
  - **Low** modality within **each cluster**

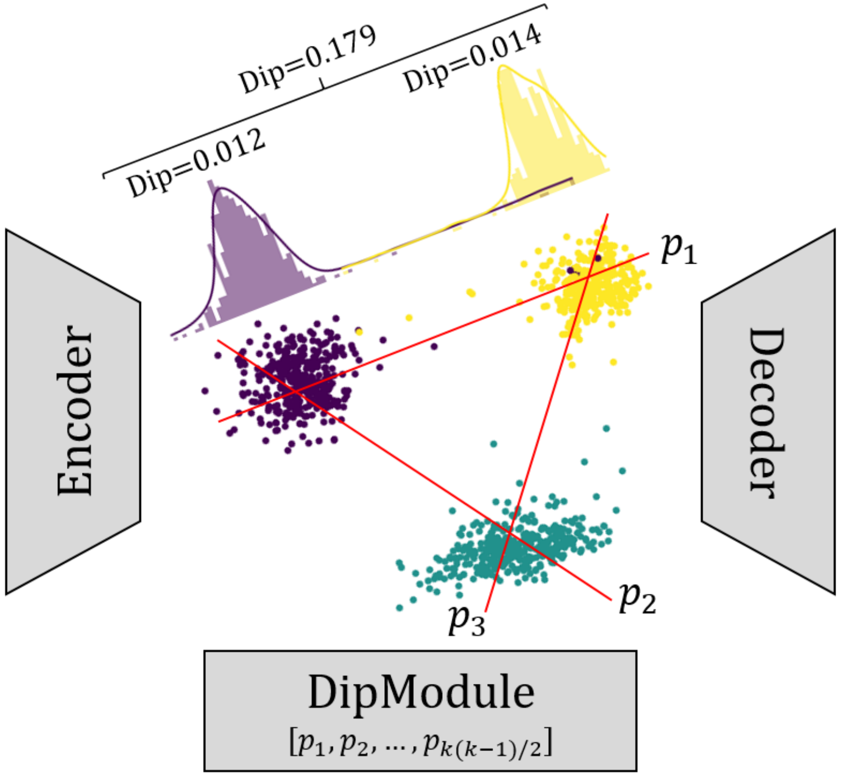


DipEncoder I

- In multidimensional space, the Dip-test is usually performed with data projected onto a projection axis
- We create one projection axis $p_{a,b}$ for each combination of clusters
- Those axes are stored in a separate NN
  → The DipModule
- The update of the autoencoder should result in
  - **High** modality between **two clusters**
  - **Low** modality within **each cluster**
- Losses:

$$\mathcal{L}_{uni}(\mathcal{B}, a, b) = \frac{1}{2}\left(\text{dip}(\overline{Z}^{\mathcal{B}}_{a,\not{b}}) + \text{dip}(\overline{Z}^{\mathcal{B}}_{\not{a},b})\right)$$

$$\mathcal{L}_{multi}(\mathcal{B}, a, b) = -\,\text{dip}(\overline{Z}^{\mathcal{B}}_{a,b})$$



DipEncoder II

- Dip loss: $\mathcal{L}_{dip}(\mathcal{B}) = \frac{2}{k(k-1)} \sum_{a=1}^{(k-1)} \sum_{b=a+1}^{k} \mathcal{L}_{uni}(\mathcal{B}, a, b) + \mathcal{L}_{multi}(\mathcal{B}, a, b)$
- Final loss: $\mathcal{L}_{final}(\mathcal{B}) = \mathcal{L}_{dip}(\mathcal{B}) + \lambda \mathcal{L}_{rec}(\mathcal{B})$

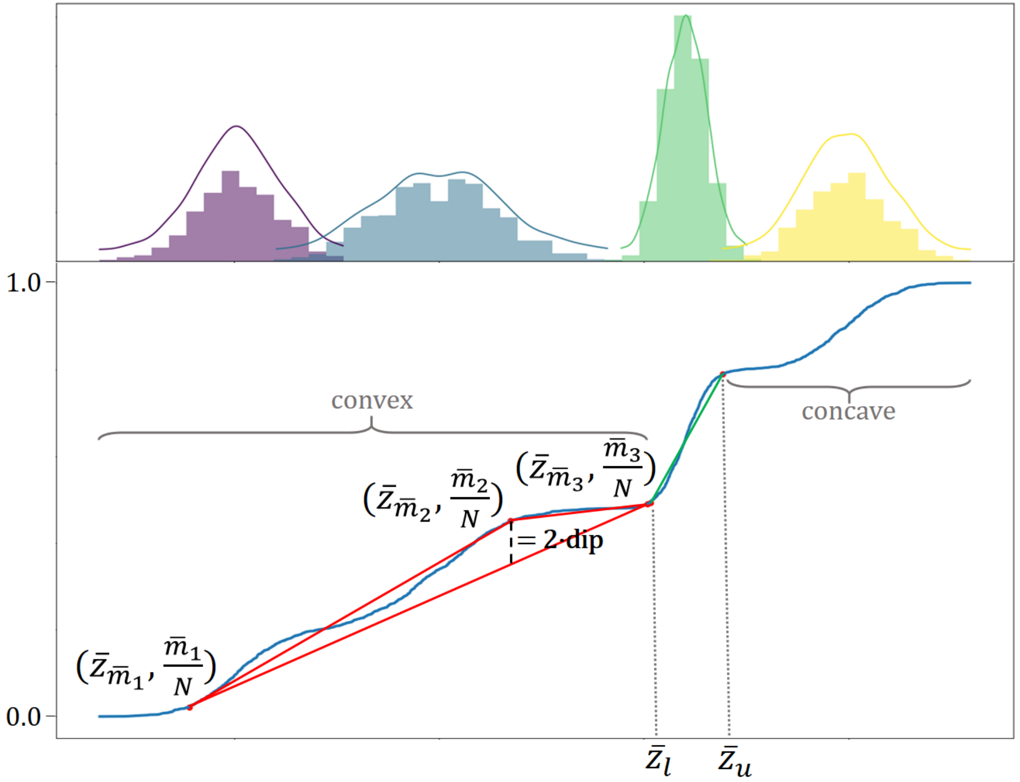
- The Dip-test can be derived to identify axes that show a high modality
  → Gradient is used to update the DipModule
- Additionally, we can derive the Dip-test with respect to the data
  → Gradient is used to update the autoencoder

Dip-test calculation

- Is calculated using the ECDF - more precisely it uses
  - The modal triangle: $\Delta = ((\overline{z}_{\overline{m}_1}, \frac{\overline{m}_1}{N}), (\overline{z}_{\overline{m}_2}, \frac{\overline{m}_2}{N}), (\overline{z}_{\overline{m}_3}, \frac{\overline{m}_3}{N}))$
  - The modal interval: $[\overline{z}_l, \overline{z}_u]$

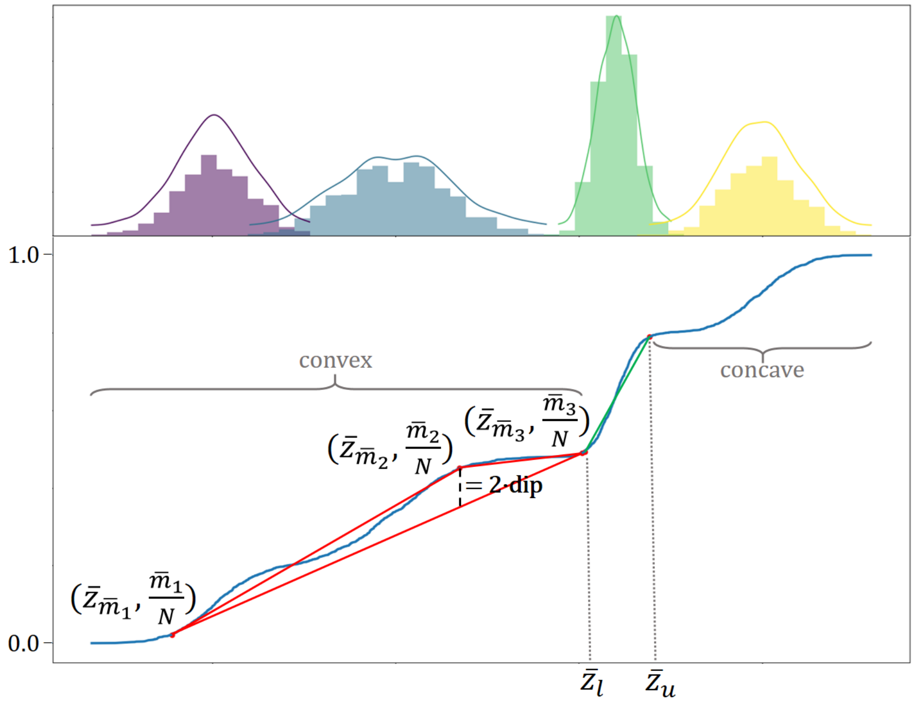
- $\overline{Z} = \mathrm{sort}\{p_{a,b}^T \cdot z | z \in \mathrm{enc}(X_{a,b})\}$

- $\mathrm{dip}(\overline{Z}) = \frac{1}{2N}\left(\left|\overbrace{\frac{(\overline{m}_3 - \overline{m}_1)(\overline{z}_{\overline{m}_2} - \overline{z}_{\overline{m}_1})}{\overline{z}_{\overline{m}_3} - \overline{z}_{\overline{m}_1}} + \overline{m}_1 - \overline{m}_2}^{=:A}\right| + 1\right)$

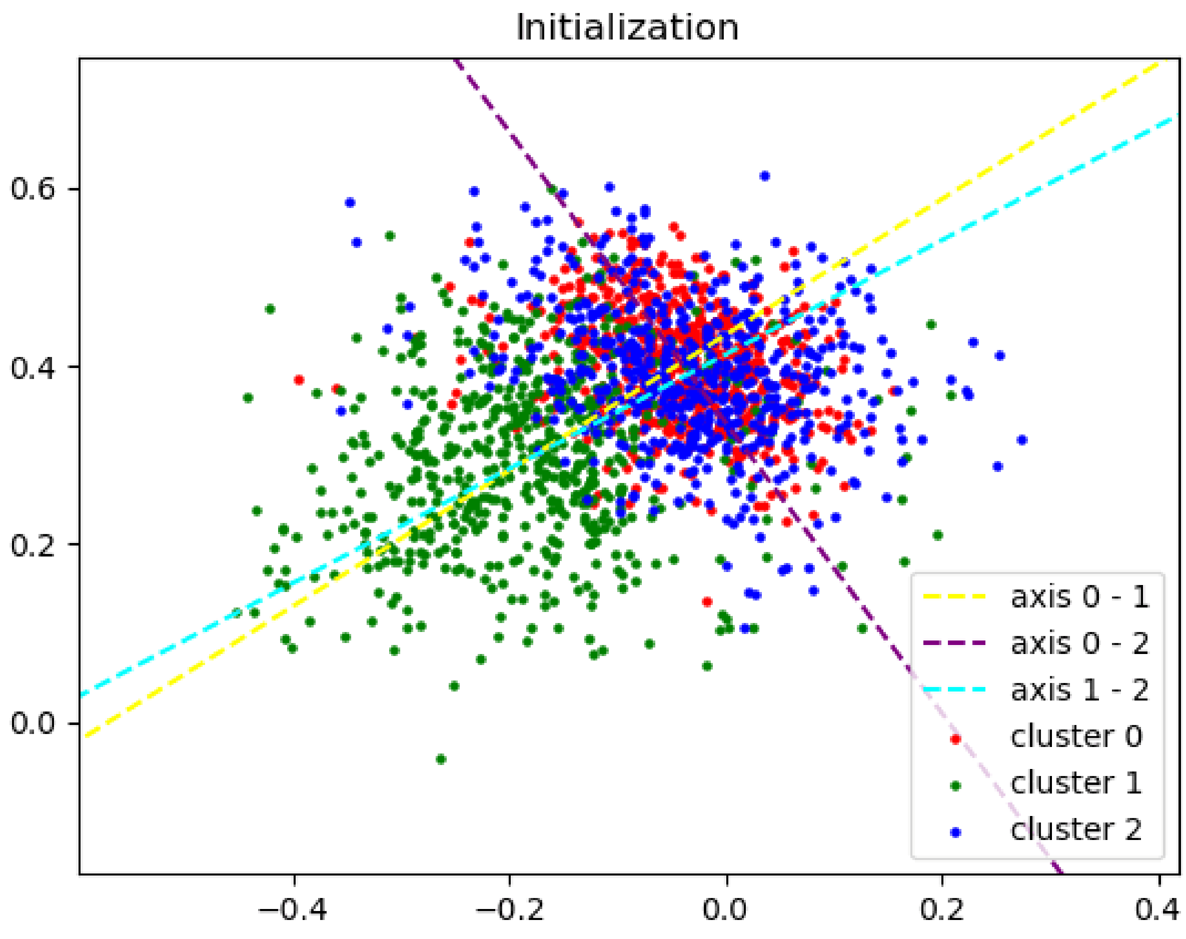


Dip-test gradients

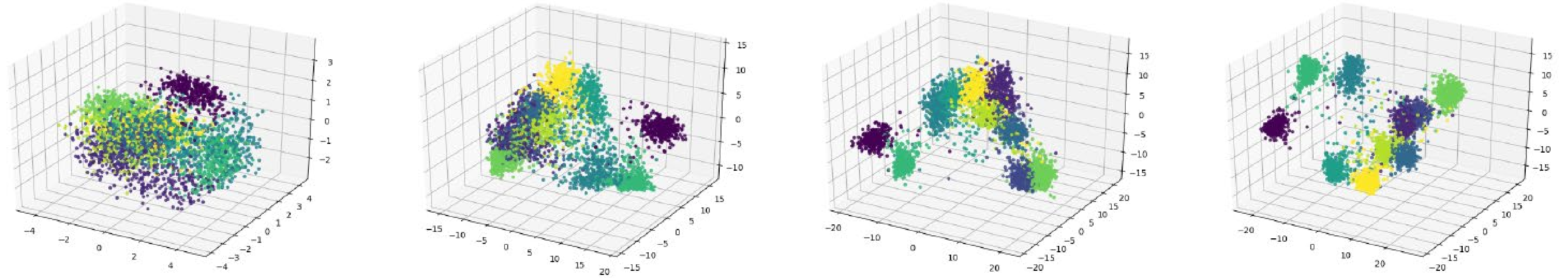
- $\overline{Z} = \text{sort}\{p_{a,b}^T \cdot z | z \in \text{enc}(X_{a,b})\}$
- $\text{dip}(\overline{Z}) = \frac{1}{2N}\left(\left|\overbrace{\frac{(\overline{m}_3 - \overline{m}_1)(\overline{z}_{\overline{m}_2} - \overline{z}_{\overline{m}_1})}{\overline{z}_{\overline{m}_3} - \overline{z}_{\overline{m}_1}} + \overline{m}_1 - \overline{m}_2}^{=: A}\right| + 1\right)$
- $\frac{\partial \text{dip}(\overline{Z})}{\partial p_{a,b}[i]} = c\left(\frac{z_{m_2}[i] - z_{m_1}[i]}{\overline{z}_{\overline{m}_3} - \overline{z}_{\overline{m}_1}} + \frac{(\overline{z}_{\overline{m}_1} - \overline{z}_{\overline{m}_2})(z_{m_3}[i] - z_{m_1}[i])}{(\overline{z}_{\overline{m}_3} - \overline{z}_{\overline{m}_1})^2}\right)$
- $\frac{\partial \text{dip}(\overline{Z})}{\partial z[i]} = \begin{cases} p_{a,b}[i] c \frac{\overline{z}_{\overline{m}_2} - \overline{z}_{\overline{m}_3}}{(\overline{z}_{\overline{m}_3} - \overline{z}_{\overline{m}_1})^2} & \text{if } z = z_{m_1}, \\ p_{a,b}[i] c \frac{1}{\overline{z}_{\overline{m}_3} - \overline{z}_{\overline{m}_1}} & \text{if } z = z_{m_2}, \\ p_{a,b}[i] c \frac{\overline{z}_{\overline{m}_1} - \overline{z}_{\overline{m}_2}}{(\overline{z}_{\overline{m}_3} - \overline{z}_{\overline{m}_1})^2} & \text{if } z = z_{m_3}, \\ 0 & \text{else}, \end{cases}$



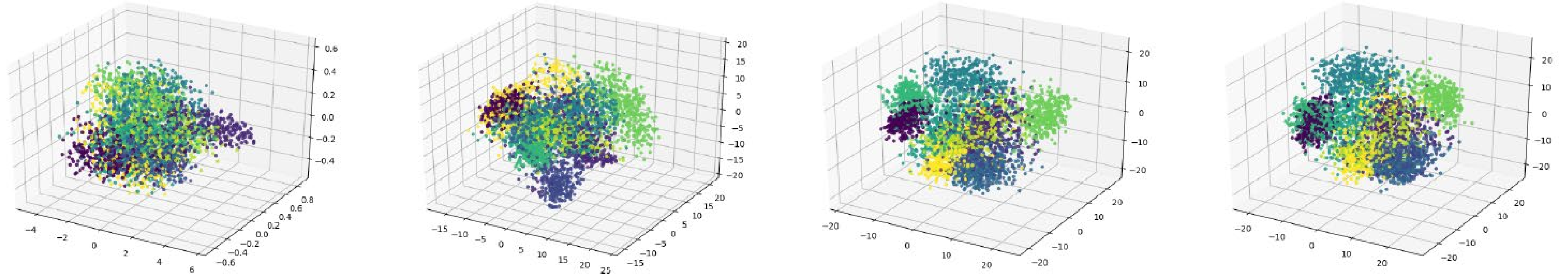
DipEncoder – Exemplary run



DipEncoder – Embedding (Optdigits)



(a) DipEncoder after 1 epoch. (b) DipEncoder after 10 epochs. (c) DipEncoder after 50 epochs. (d) DipEncoder after 100 epochs.

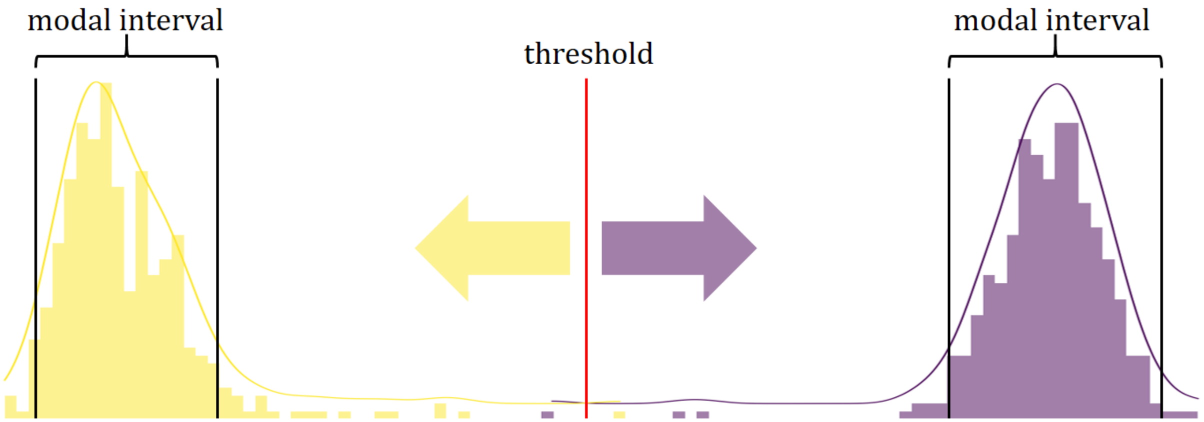


(e) AE after 1 epoch. (f) AE after 10 epochs. (g) AE after 50 epochs. (h) AE after 100 epochs.

* In this experiment we used the ground truth labels to create the embedding

Update the cluster labels

- The modal interval, a byproduct of the Dip-test, can be interpreted as the main data range of a cluster
- Based on the center point between two clusters on $p_{a,b}$ we decide if the right or left cluster is a better fit



- We check this for each combination of clusters, respectively for each $p_{a,b}$, and finally choose the label of the cluster that matched most often

Deep Clustering algorithm

- Pretrain regular autoencoder
- Execute k-means
- Initialize the DipModule using the k-means centers
- In each epoch do:
  - Update labels using current projection axes
  - Update the DipModule and the autoencoder using $\mathcal{L}_{final}(\mathcal{B}) = \mathcal{L}_{dip}(\mathcal{B}) + \lambda\mathcal{L}_{rec}(\mathcal{B})$

Algorithm 1: Pseudocode of the DipEncoder

Input: data set X, number of clusters k, number of epochs E
Output: *labels*

// Pretrain AE; save the reconstruction loss of $\mathcal{B}_{init}$ as λ
(AE, λ) = pretrain autoencoder on X using $\mathcal{L}_{rec}$
// Get initial labels and projection axes
$labels$ = k-means(AE.encode(X), k)
DM = DipModule($X, AE, labels$)
for $epoch = 0;\ epoch \leq E;\ epoch \mathrel{+}= 1$ **do**
    // Update *labels* as described
    for $x \in AE.encode(X)$ **do**
        $clusterMatches = [0, \ldots, 0] \in \mathbb{R}^k$
        for $a = 1;\ a \leq k-1;\ a \mathrel{+}= 1$ **do**
            for $b = a+1;\ b \leq k;\ b \mathrel{+}= 1$ **do**
                $p_{a,b} = DM.\text{getProjectionAxis}(a, b)$
                $\overline{Z}_{a,\cancel{b}} = \text{sort}\{p_{a,b}^T \cdot z | z \in AE.\text{encode}(X_a)\}$
                $\overline{Z}_{\cancel{a},b} = \text{sort}\{p_{a,b}^T \cdot z | z \in AE.\text{encode}(X_b)\}$
                $[\overline{z}_{l,a}, \overline{z}_{u,a}], [\overline{z}_{l,b}, \overline{z}_{u,b}] = \text{dip}(\overline{Z}_{a,\cancel{b}}), \text{dip}(\overline{Z}_{\cancel{a},b})$
                c_L, c_R = ids of left and right cluster on $p_{a,b}$
                T = center between the clusters
                if $(p_{a,b}^T \cdot x) < T$ **then**
                    $clusterMatches[c_L] \mathrel{+}= 1$
                else
                    $clusterMatches[c_R] \mathrel{+}= 1$
        set label of x to argmax($clusterMatches$)
    if $epoch == E$ **then**
        break
    // Train the DipEncoder
    for $\mathcal{B}$ ***in*** X **do**
        $\mathcal{L}_{final} = \mathcal{L}_{dip}(\mathcal{B}) + \lambda\mathcal{L}_{rec}(\mathcal{B})$
        optimize AE and DM using $\mathcal{L}_{final}$
return *labels*

DipEncoder vs. other dimensionality reduction techniques

- We compare the DipEncoder to different dimensionality reduction techniques in combination with SVM
- Therefore, we use the ground truth labels of the training data to train the models and predict the labels of the test data
- The common test/train split of MNIST is used

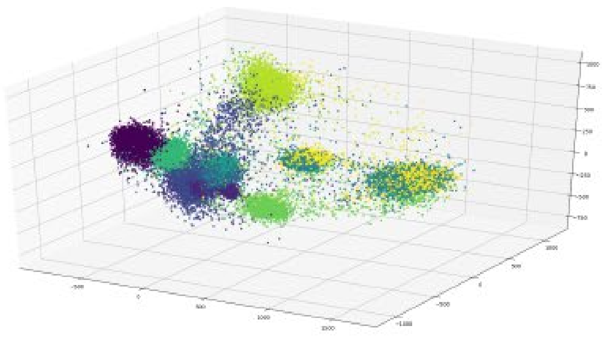
Method	**MNIST** ($k=10$) $\left(\begin{smallmatrix} N_{\text{train}} = 60000 \\ N_{\text{test}} = 10000 \end{smallmatrix}, d = 784\right)$	
	ACC	NMI
DipEncoder$_{supervised}$	$\underline{94.2} \pm 3.9$ ($\underline{97.2}$)	$\underline{90.5} \pm 2.3$ ($\underline{92.7}$)
DipEncoder+SVM	92.9 ± 4.3 (97.1)	88.1 ± 3.1 (92.5)
SVM	86.6 ± 1.1 (87.5)	74.5 ± 1.1 (75.5)
PCA+SVM	58.5 ± 4.9 (67.7)	45.9 ± 3.4 (53.9)
LDA+SVM	87.7 ± 0.0 (87.7)	74.7 ± 0.0 (74.7)
AE+SVM	89.1 ± 1.7 (91.2)	79.4 ± 2.0 (81.8)

NMI results

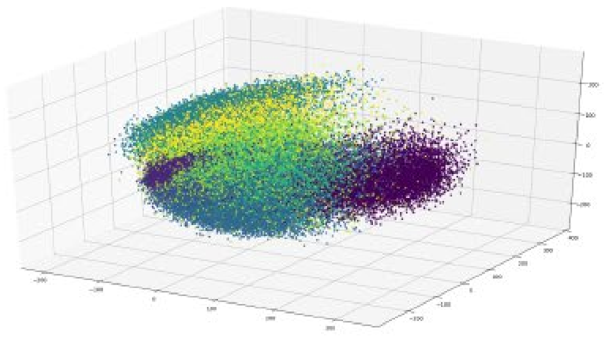
Method	Optdigits (k = 10) (N = 5620, d = 64)	USPS (k = 10) (N = 9298, d = 256)	HAR (k = 6) (N = 10299, d = 561)	Pendigits (k = 10) (N = 10992, d = 16)	Reuters10k (k = 4) (N = 10000, d = 2000)
DipEncoder	88.6 ± 3.0 (92.0)	81.9 ± 0.8 (83.6)	73.5 ± 6.9 (82.4)	75.2 ± 2.1 (78.2)	36.8 ± 4.2 (41.9)
AE+k-means	80.1 ± 2.4 (83.8)	69.6 ± 0.9 (71.2)	67.5 ± 4.2 (73.2)	70.0 ± 0.8 (72.2)	37.2 ± 6.3 (47.3)
DEC	88.5 ± 2.5 (91.9)	80.7 ± 0.6 (81.4)	66.3 ± 4.8 (76.8)	76.9 ± 1.1 (77.9)	37.9 ± 7.1 (51.6)
IDEC	80.4 ± 2.4 (84.0)	69.3 ± 1.0 (70.9)	69.9 ± 2.9 (74.1)	69.8 ± 1.3 (72.2)	39.1 ± 6.7 (51.9)
DCN	84.8 ± 2.3 (84.8)	74.6 ± 1.3 (76.1)	73.4 ± 4.8 (80.9)	73.5 ± 0.5 (74.4)	35.1 ± 7.1 (45.4)
DipDECK	83.5 ± 2.3 (86.7)	68.5 ± 1.3 (70.5)	70.8 ± 1.3 (72.1)	72.8 ± 1.2 (74.7)	15.6 ± 18.1 (45.8)

Method	20Newsgroups (k = 20) (N = 18846, d = 2000)	Letters (k = 26) (N = 20000, d = 16)	MNIST (k = 10) (N = 70000, d = 784)	F-MNIST (k = 10) (N = 70000, d = 784)	K-MNIST (k = 10) (N = 70000, d = 784)
DipEncoder	30.8 ± 0.6 (31.6)	47.1 ± 0.9 (48.2)	85.8 ± 1.6 (87.8)	60.6 ± 2.2 (63.5)	52.2 ± 3.2 (57.0)
AE+k-means	31.2 ± 0.8 (32.4)	42.3 ± 0.9 (43.4)	74.4 ± 1.5 (77.0)	54.2 ± 0.5 (55.1)	46.3 ± 2.4 (49.7)
DEC	15.8 ± 1.0 (17.1)	46.0 ± 2.0 (48.0)	85.2 ± 1.2 (86.6)	59.7 ± 1.4 (62.5)	54.2 ± 2.1 (58.0)
IDEC	28.1 ± 1.3 (30.2)	45.1 ± 1.4 (47.6)	75.3 ± 1.2 (77.8)	54.3 ± 0.7 (55.3)	46.7 ± 1.8 (49.1)
DCN	28.6 ± 1.5 (30.7)	43.7 ± 2.4 (48.4)	82.0 ± 1.7 (84.2)	56.0 ± 0.9 (58.3)	48.2 ± 2.0 (51.7)
DipDECK	00.1 ± 0.0 (00.1)	34.5 ± 3.6 (38.2)	75.8 ± 2.0 (79.4)	53.9 ± 2.9 (57.2)	38.7 ± 4.1 (43.0)

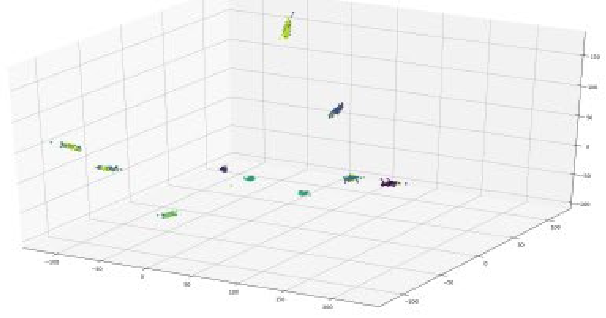
Deep Clustering - Embedding (MNIST)



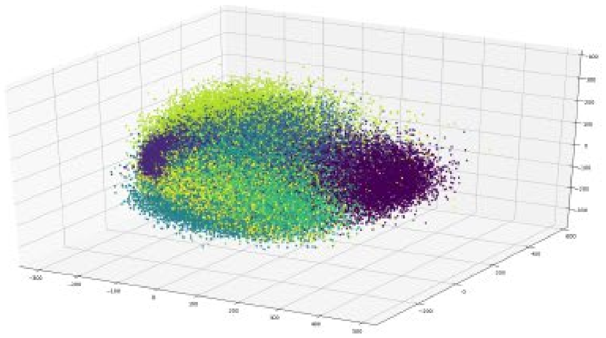
(a) DipEncoder.



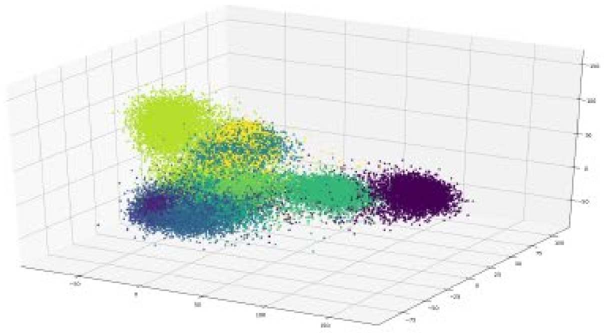
(b) AE+k-means.



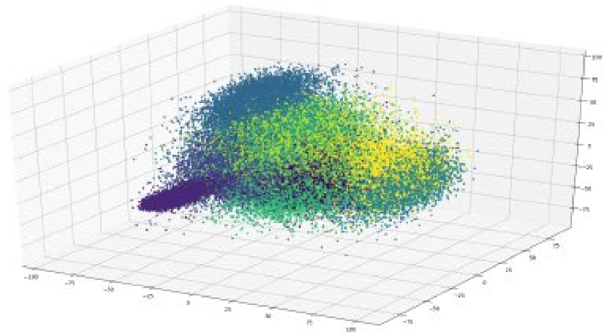
(c) DEC.



(d) IDEC.



(e) DCN.



(f) DipDECK.

Conclusion

- We successfully combined the previously unused gradient of the Dip-value with respect to the data with an autoencoder to create cluster-friendly embeddings
- No underlying distribution functions are necessary

- Based on this, we have created a novel Deep Clustering algorithm that is solely based on the Dip-test
- Experiments show that the DipEncoder produces superior results compared to competitor algorithms

Thank you for your attention!

Contact: leiber@dbs.ifi.lmu.de

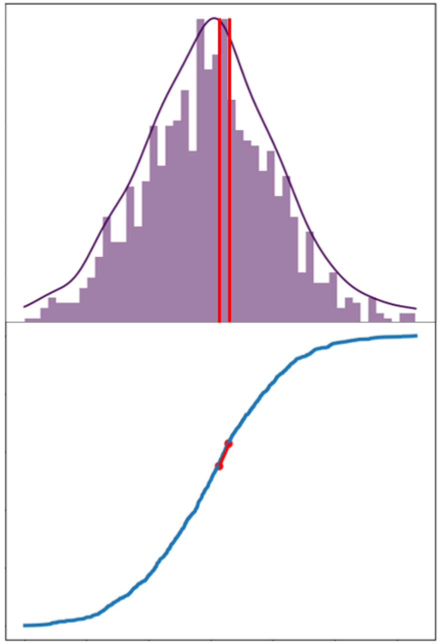
ARI results

Method	Optdigits ($k = 10$) ($N = 5620, d = 64$)	USPS ($k = 10$) ($N = 9298, d = 256$)	HAR ($k = 6$) ($N = 10299, d = 561$)	Pendigits ($k = 10$) ($N = 10992, d = 16$)	Reuters10k ($k = 4$) ($N = 10000, d = 2000$)
DipEncoder	85.6 ± 5.8 (91.4)	74.2 ± 1.1 (76.2)	63.4 ± 7.4 (73.5)	64.7 ± 4.2 (70.2)	35.4 ± 4.5 (43.4)
AE+k-means	76.7 ± 4.5 (83.0)	59.7 ± 1.4 (61.6)	58.7 ± 5.4 (65.1)	60.5 ± 1.9 (63.9)	39.9 ± 10.7 (56.7)
DEC	84.9 ± 5.2 (91.1)	72.7 ± 0.9 (74.1)	53.4 ± 7.7 (71.8)	66.7 ± 2.6 (68.6)	35.3 ± 10.6 (57.6)
IDEC	77.1 ± 4.6 (83.1)	59.3 ± 1.3 (60.9)	58.1 ± 2.8 (62.5)	60.2 ± 2.9 (63.9)	36.4 ± 10.1 (56.4)
DCN	81.1 ± 5.0 (86.2)	64.4 ± 2.3 (67.0)	62.7 ± 5.8 (71.9)	62.6 ± 2.0 (64.7)	29.6 ± 10.1 (49.8)
DipDECK	79.6 ± 5.5 (85.7)	58.7 ± 2.8 (63.1)	49.9 ± 1.1 (50.9)	61.7 ± 2.6 (65.4)	12.1 ± 14.3 (36.9)

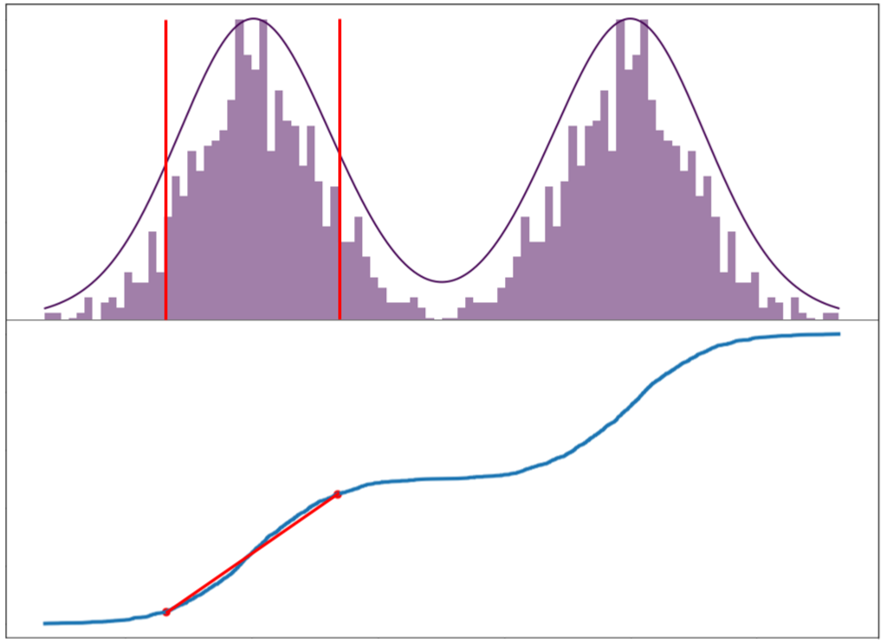
Method	20Newsgroups ($k = 20$) ($N = 18846, d = 2000$)	Letters ($k = 26$) ($N = 20000, d = 16$)	MNIST ($k = 10$) ($N = 70000, d = 784$)	F-MNIST ($k = 10$) ($N = 70000, d = 784$)	K-MNIST ($k = 10$) ($N = 70000, d = 784$)
DipEncoder	18.0 ± 0.9 (19.2)	22.8 ± 0.9 (24.0)	81.0 ± 3.0 (84.5)	44.8 ± 2.8 (47.4)	37.7 ± 3.7 (43.9)
AE+k-means	16.9 ± 0.7 (17.9)	18.8 ± 0.6 (19.9)	69.1 ± 2.3 (73.2)	38.3 ± 1.1 (39.9)	32.4 ± 3.1 (37.9)
DEC	5.4 ± 0.6 (6.1)	20.5 ± 2.3 (23.4)	81.6 ± 2.1 (83.5)	44.0 ± 2.8 (48.9)	39.0 ± 2.3 (42.3)
IDEC	12.8 ± 1.1 (14.8)	21.3 ± 1.5 (23.6)	70.3 ± 2.0 (74.2)	38.1 ± 1.1 (39.9)	32.5 ± 2.2 (36.5)
DCN	15.1 ± 1.1 (17.4)	18.9 ± 2.5 (23.9)	77.1 ± 3.5 (80.8)	38.5 ± 1.2 (41.3)	31.5 ± 2.9 (37.5)
DipDECK	00.0 ± 0.0 (00.0)	7.2 ± 1.8 (9.1)	70.7 ± 2.5 (74.4)	32.8 ± 3.3 (37.6)	22.1 ± 4.0 (29.0)

Problem with the modal interval

- For a single unimodal structure the modal interval gets very small
  → Problem for the update of the cluster labels



- Solution: Mirror the dataset



Influence of the batch size

- The Dip-test only returns meaningful values if a certain amount of samples is present
  - → We need larger batch sizes with more clusters present
  - → We recommend a batch size of $25 \cdot k$