Guided Clustering Variational Autoencoder

Violaine Courrier, Christophe Biernacki

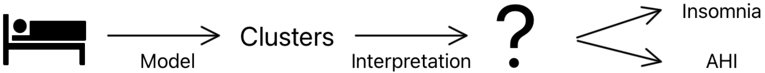
Withings & Inria Lille

Workshop Fondements Mathématiques de l'IA

March 25, 2025

Context-aware clustering

Lack of context-awareness: same variables, different goals.



Strategies :

- Variable selection: determine which features are relevant to each context.
- Feature weighting: assign different weights based on their importance to a context.
- Constraint-based clustering: "these points must be in the same cluster".
- Add a context variable: add an extra dimension that describes the context.

Guiding variable

To respond to the problem, we want clusters that are generative of the input variables but also a **guiding variable** y.

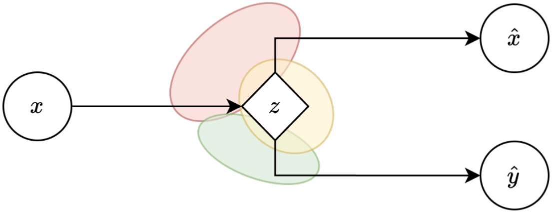


Figure: Illustration of the model. *Diamond-shaped nodes denote latent variables, while round ones denote observations.*

Generative model

- Choose a cluster c: $p_{\boldsymbol{\pi}}(c) = Cat(c; \boldsymbol{\pi})$.
- Generate a latent vector z conditioned on the cluster c: $p_{\mu_c,\sigma_c}(z|c) = \mathcal{N}(z; \mu_c, \sigma_c^2 I)$.

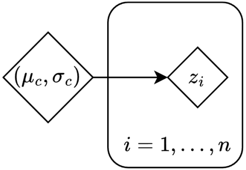


Figure: A graphical representation of the GMM.

Here $\boldsymbol{\pi} = (\pi_1, ..., \pi_K) \in [0, 1]^K$, $\sum_{c=1}^{K} \pi_c = 1$, μ_c and σ_c^2 the mean and the diagonal covariance of the multivariate normal distribution corresponding to cluster c.

Generative model

- Generate the variable x from the latent vector z: $p_{\theta_x}(x|z) = \mathcal{N}(x; f_{\theta_x}(z), I)$.
- Generate the variable y from the latent vector z: $p_{\theta_y}(y|z) = \mathcal{N}(y; f_{\theta_y}(z), I)$.

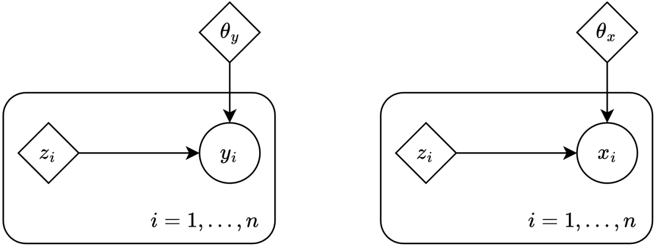


Figure: A graphical representation of the decoders.

Here I is the identity matrix, and $f_{\theta_a}(z)$ with $a \in \{x, y\}$ are networks with input z and parametrized by θ_a.

Variational inference

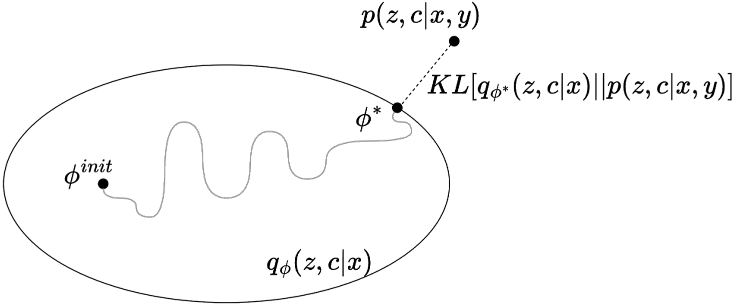


Figure: Illustration inspired by [Blei et al.,].

- Fit the variational parameters ϕ to be close in KL to the exact posterior.

Inference model

- Mean-field approximation: $q_\phi(z, c|x) = q_\phi(z|x)q(c|x)$, with

$$q_\phi(z|x) = \mathcal{N}(z; \tilde{\mu}, \tilde{\sigma}^2 I)$$
$$[\tilde{\mu}, \log \tilde{\sigma}] = g_\phi(x)$$

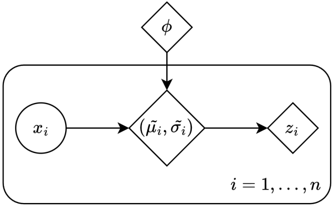


Figure: A graphical representation of the inference model.

Objective

$$\mathsf{ELBO}(x,y) = \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\Theta(y|z)]}_{\text{Reconstruction of } y} + \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\Theta(x|z)]}_{\text{Reconstruction of } x} - \underbrace{\mathsf{KL}[q_\phi(z,c|x)||p(z|c)p(c)]}_{\text{Structure latent space}}$$

Example 1: Withings' sleep dataset guided by AHI

- In x, data available with the watch and the sleep analyzer
  - Sleep duration ($+$ light/deep sleep duration)
  - Number of sleep interruptions
  - BMI
  - Age
- In y, the AHI score, available only with the sleep analyzer

$\Rightarrow$ Goal: Find clusters that are interpretable in the sense of their AHI score without using the information as input.

Impact of the guiding variable

- Comparison of the clustering with and without the AHI as a guiding variable.

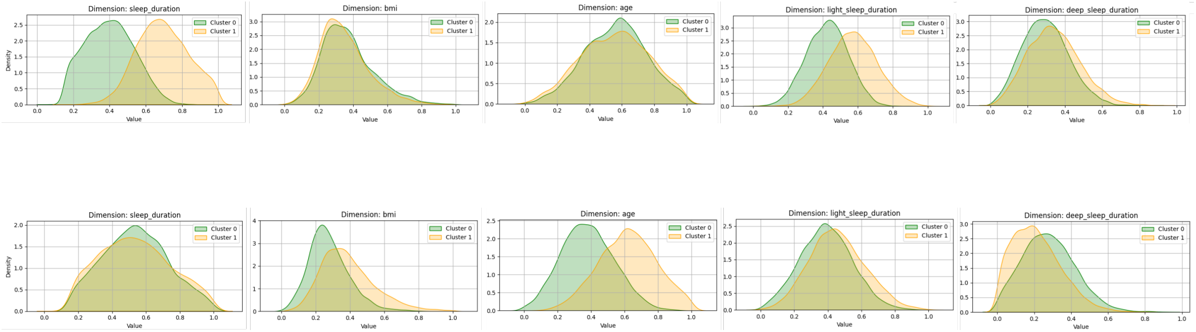
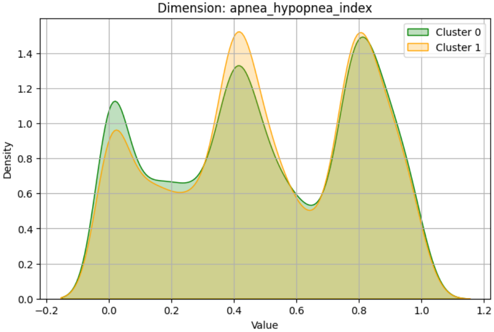


Figure: Kernel density plots comparing per-feature distributions across the two clusters in the test dataset. (Up) Model without any guiding variable. (Down) Model with AHI as the guiding variable.

Impact of the guiding variable



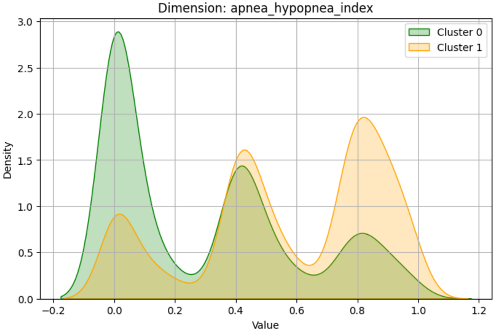


Figure: Kernel density plot illustrating the distribution of AHI values across each cluster in the test dataset. (Left) Model without any guiding variable. (Right) Model with AHI as the guiding variable.

Contribution

- **Contextually guided clustering**: introduction of a guiding variable y, preserving the full richness of the original dataset x.
- **Adaptability to different contexts**: adapt to different contexts by changing the guiding variable y, allowing for flexibility to adjust the clustering objective to a new context.
- **Generative architecture for interpretability**: can generate both input data x and guiding variables y along with the clustering, enhancing cluster interpretability.
- **Inference independence**: y is used only in the generative model, leaving the inference process to rely solely on x. The model remains applicable even when y is unavailable at prediction time.
- **Uncertainty quantification**: leverages the VAE's probabilistic nature to estimate cluster membership probabilities and quantify assignment uncertainties.

References I

Blei, D., Ranganath, R., and Mohamed, S.
Variational Inference: Foundations and Modern Methods.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977).
Maximum Likelihood from Incomplete Data Via the *EM* Algorithm.
Journal of the Royal Statistical Society Series B: Statistical Methodology, 39(1):1–22.

Falck, F., Zhang, H., Willetts, M., Nicholson, G., Yau, C., and Holmes, C. (2021).
Multi-Facet Clustering Variational Autoencoders.
arXiv:2106.05241 [stat].

Hu, W., Miyato, T., Tokui, S., Matsumoto, E., and Sugiyama, M. (2017).
Learning Discrete Representations via Information Maximizing Self-Augmented Training.
arXiv:1702.08720 [stat].

References II

Jiang, Z., Zheng, Y., Tan, H., Tang, B., and Zhou, H. (2017).
Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering.
arXiv:1611.05148 [cs].

Kilinc, O. and Uysal, I. (2018).
Learning Latent Representations in Neural Networks for Clustering through Pseudo Supervision and Graph-based Activity Regularization.
arXiv:1802.03063 [cs].

Kosiorek, A. R., Sabour, S., Teh, Y. W., and Hinton, G. E. (2019).
Stacked Capsule Autoencoders.
arXiv:1906.06818 [stat].

MacQueen, J. (1967).
Some methods for classification and analysis of multivariate observations.
In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5.1, pages 281–298. University of California Press.

References III

Monnier, T., Groueix, T., and Aubry, M. (2020).
Deep Transformation-Invariant Clustering.
arXiv:2006.11132 [cs].

Xie, J., Girshick, R., and Farhadi, A. (2016).
Unsupervised Deep Embedding for Clustering Analysis.
arXiv:1511.06335 [cs].

Thank you !

Probabilistic ML

- Probabilistic model is a joint distribution of hidden variables (z, c) and observed variables (x, y):
$$p_\Theta(z, c, x, y)$$
- Inference about the unknown is performed through the posterior:
$$p(z, c|x, y) = \frac{p_\Theta(z, c, x, y)}{p(x, y)}$$
- Denominator not tractable $\rightarrow$ approximate posterior inference

Objective

The ELBO is a lower bound of the observed likelihood:

$$
\begin{aligned}
&\mathsf{KL}[q_\phi(z, c|x) || p_\Theta(z, c|x, y)] \\
&\quad = -\mathbb{E}_{q_\phi(z,c|x)}\left[\log \frac{p_\Theta(z, c|x, y)}{q_\phi(z, c|x)}\right] \\
&\quad = -\mathbb{E}_{q_\phi(z,c|x)}\left[\log \frac{p_\Theta(z, c, x, y)}{q_\phi(z, c|x)}\right] + \log p(x, y) \\
&\Rightarrow \log p(x, y) \geq \mathbb{E}_{q_\phi(z,c|x)}\left[\log \frac{p_\Theta(z, c, x, y)}{q_\phi(z, c|x)}\right].
\end{aligned}
$$

The objective is to maximize the ELBO:

$$
\begin{aligned}
&\arg\max_{\Theta,\phi} \mathsf{ELBO}(x, y) \\
&\quad = \arg\max_{\Theta,\phi} \mathbb{E}_{q_\phi(z,c|x)}\left[\log \frac{p_\Theta(z, c, x, y)}{q_\phi(z, c|x)}\right].
\end{aligned}
$$

Approximate $q(c|x)$

$$
\begin{aligned}
\mathsf{ELBO}(x,y) &= \mathbb{E}_{q_\phi(z,c|x)}\left[\log \frac{p_\Theta(z,c,x,y)}{q_\phi(z,c|x)}\right] \\
&= \int \sum_{c=1}^{K} q_\phi(z|x)q(c|x) \log \frac{p_\Theta(y|z)p_\Theta(x|z)p_\Theta(z|c)p_\Theta(c)}{q_\phi(z|x)q(c|x)} dz \\
&= \int q_\phi(z|x) \log \frac{p_\Theta(y|z)p_\Theta(x|z)p_\Theta(z)}{q_\phi(z|x)} dz - \int q_\phi(z|x)\ \mathsf{KL}[q(c|x)||p_\Theta(c|z)] dz
\end{aligned}
$$

The 1st term does not depend on c and the 2nd term is non-negative.
$\Rightarrow$ Maximizing the lower bound ELBO with respect to $q(c|x)$ requires that $\mathsf{KL}[q(c|x)||p_\Theta(c|z)] = 0$. With ν a constant, we have:

$$
\frac{q(c|x)}{p_\Theta(c|z)} = \nu.
$$

Approximate $q(c|x)$

Since $\sum_c q(c|x) = 1$ and $\sum_c p_\Theta(c|z) = 1$, we have:

$$\frac{q(c|x)}{p_\Theta(c|z)} = 1.$$

Taking the expectation on both sides, we can obtain:

$$q(c|x) = \mathbb{E}_{q_\phi(z|x)}[p_\Theta(c|z)].$$

We will approximate $q(c|x)$ using the SGVB estimator:

$$q(c|x) = \mathbb{E}_{q_\phi(z|x)}[p(c|z)] \simeq \frac{1}{L}\sum_{l=1}^{L} \frac{p_\Theta(z^{(l)}|c)p_\Theta(c)}{\sum_{c'} p_\Theta(z^{(l)}|c')p_\Theta(c')}.$$

Developed ELBO

$$
\begin{aligned}
\mathsf{ELBO}(x,y) = &-\frac{1}{L}\sum_{l=1}^{L}||y - f_{\theta_y}(z^{(l)})||_2^2 - \frac{1}{L}\sum_{l=1}^{L}||x - f_{\theta_x}(z^{(l)})||_2^2 \\
&- \frac{1}{2}\sum_{c=1}^{K} q(c|x)\sum_{j=1}^{J}\left(\log\sigma_{cj}^2 + \frac{\tilde{\sigma}_j^2}{\sigma_{cj}^2} + \frac{(\tilde{\mu}_j - \mu_{cj})^2}{\sigma_{cj}^2}\right) \\
&+ \sum_{c=1}^{K} q(c|x)\log\pi_c + \frac{1}{2}\sum_{j=1}^{J}(1+\log\tilde{\sigma}_j^2) - \sum_{c=1}^{K} q(c|x)\log q(c|x)
\end{aligned}
$$

with $z^{(l)} \sim \mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2 I)$, and $[\tilde{\mu}, \log\tilde{\sigma}^2] = g_\phi(x)$. L is the number of Monte Carlo samples in the Stochastic Gradient Variational Bayes estimator and J is the dimension of z.

Initialization GMM

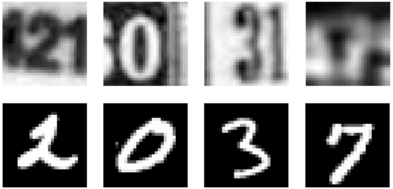
- Pre-training is used to initialize GMM parameters (μ, σ).
- We introduce the weight α in the loss function to balance the reconstruction of x, y and the structure of the latent space without the clusters

$$\begin{aligned}\mathsf{ELBO}_{\alpha,\beta}(x,y) = \alpha \times \mathbb{E}_{q_\phi(z|x)}[\log p_\Theta(y|z)] + \mathbb{E}_{q_\phi(z|x)}[\log p_\Theta(x|z)] \\ - \beta \times \mathsf{KL}[q_\phi(z|x) || \mathcal{N}(z; 0_J, I)]\end{aligned}$$

with 0_J a vector null of dimension J.

- By prioritizing the reconstruction of y with $\alpha > 1$, we encourage the model to align the latent space with the guiding variable.

Example 2: MNIST/SVHN



- x: SVHN images
- y: MNIST images

$\Rightarrow$ Goal: Find clusters that are generative both of SVHN and MNIST images, therefore that have a meaning in both domains.

Generative property of the clusters

- The generative aspect - key for the interpretation

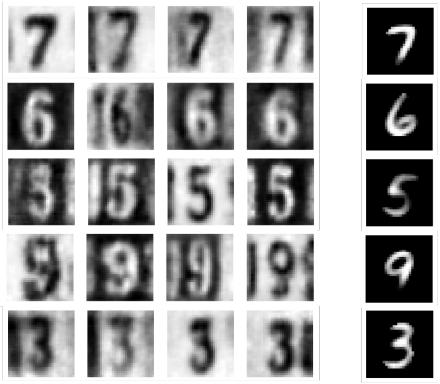


Figure: Four examples of generated SVHN images (left) and a generated MNIST images (right) of five clusters.

Implementation details MNIST/SVHN

- CNNs for both the encoder and the decoder of SVHN, and a MLP for the decoder of MNIST.
- Latent space dimension: 20.
- $\beta = 3$.
- Pretraining: $\alpha = 10$, learning rate $= 0.0001$.
- Training: learning rate of 0.001 for the parameters of the encoder and decoders, and 0.0001 for the parameters of the GMM, and set the number of clusters to 10.

Visualization of the clusters during the training

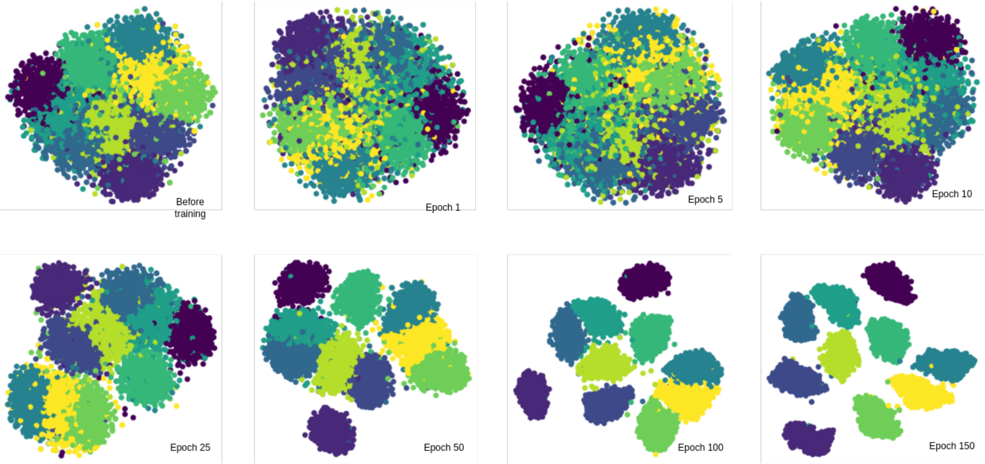


Figure: t-SNE visualisations at different epoch during the training of $5,000$ training images.

Comparison of the performance on standard benchmarks

Table: ACC on standard clustering benchmarks.

Model	ACC
Clustering models with image-specific transformations	
DTI K-means [Monnier et al., 2020]	44.5%
SCAE [Kosiorek et al., 2019]	55.3%
DTI GMM [Monnier et al., 2020]	57.4%
ACOL-GAR [Kilinc and Uysal, 2018]	**76.8%**
Clustering models with domain-agnostic designs	
GMM [Dempster et al., 1977]	11.6%
DEC [Xie et al., 2016]	11.9%
K-means [MacQueen, 1967]	12.2%
VaDE [Jiang et al., 2017]	30.8%
MFCVAE [Falck et al., 2021]	56.3%
IMSAT [Hu et al., 2017]	57.3%
GCVAE	**64.2%**

Implementation details Withings sleep dataset

- MLPs in the encoder and both the decoders.
- Latent space dimension: 5.
- $\beta = 0.03$.
- Pretraining: $\alpha = 10$, learning rate $= 0.00005$.
- Training: learning rate of 0.0001 for the parameters of the encoder and decoders, and 0.00001 for the parameters of the GMM, and set the number of clusters to 2.

Withings' dataset details

- 50,000 individuals - 1 night per individual
- Recorded by the Withings Sleep Analyzer[1]
- Equal number of users across the three categories based on the AHI

Table: Descriptive statistics of the variables

Variable (unit)	**Range**	**Mean**	**Std. Dev.**
sleep_duration (seconds)	14880 – 36000	26224	4224
light_sleep_duration (seconds)	3600 – 31860	15747	4651
deep_sleep_duration (seconds)	3600 – 32220	10472	4022
nb_sleep_interruptions	0 – 20	2.74	2.34
avg_night_hr (bpm)	40 – 111	62.49	8.57
bmi (kg/m²)	16 – 50	27.53	5.14
age (years)	18 – 80	50	12.67
apnea_hypopnea_index	0 – 40	18.14	13.47

[1] https://www.withings.com/us/en/sleep

Dynamic clustering: Extend the model for time series data to allow individual cluster assignments to evolve over time.