Deep Clustering via Joint Convolutional Autoencoder Embedding and Relative Entropy Minimization

Kamran Ghasedi Dizaji†, Amirhossein Herandi‡, Cheng Deng♯, Weidong Cai♮, Heng Huang†

†Department of Electrical and Computer Engineering, University of Pittsburgh, USA ‡Department of Computer Science and Engineering, University of Texas at Arlington, USA

♯School of Electronic Engineering, Xidian University, China ♮School of Information Technologies, University of Sydney, Australia

kag221@pitt.edu, amirhossein.herandi@uta.edu, chdeng@mail.xidian.edu.cn, tom.cai@sydney.edu.au, heng.huang@pitt.edu

Motivations

Dealing with large-size and high-dimensional data, existing clustering algorithms suffer from different issues, such as using inflexible hand-crafted features, shallow and linear embedding functions, non-joint embedding and clustering processes and complicated clustering algorithms that require tuning hyper-parameters.

In this paper, we propose a new clustering model, called **DE**e**P** Embedded Regular**I**zed **C**lus**T**ering (*DEPICT*), which adrress these issues by efficiently maping data into a discriminative embedding subspace and precisely predicting cluster assignments. *DEPICT* generally

- consists of a multinomial logistic regression function stacked on top of a multi-layer convolutional autoencoder.
- employs a clustering objective function using relative entropy (*KL* divergence) minimization, regularized by a prior for the frequency of cluster assignments.
- utilizes a joint learning framework to benefit from end-to-end optimization and eliminate the necessity for layer-wise pretraining, by minimizing the unified clustering and reconstruction loss functions together and trains all network layers simultaneously.

Proposed Model and Algorithm

Notations: Let's consider the clustering task of N samples, $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]$, into K categories, where each sample $\mathbf{x}_i \in \mathbb{R}^{d_x}$. Using an embedding function, we are able to map raw samples into the embedding subspace $\mathbf{Z} = [\mathbf{z}_1, ..., \mathbf{z}_n]$, where each $\mathbf{z}_i \in \mathbb{R}^{d_z}$ and $d_z \ll d_x$. Given the embedded features, we use a multinomial logistic regression (soft-max) function $p_{ik} = P(y_i = k|\mathbf{z}_i, \mathbf{\Theta}) \propto exp(\boldsymbol{\theta}_k^T \mathbf{z}_i)$ to predict the probabilistic cluster assignments.

Objective Function: In order to define our clustering objective function, we employ an auxiliary target variable $\mathbf{Q}$ to refine the model predictions iteratively. To do so, we use Kullback-Leibler (KL) divergence $KL(\mathbf{Q}\|\mathbf{P})$ to decrease the distance between the model predictions $\mathbf{P}$ and the target variables $\mathbf{Q}$.

To avoid degenerate solutions, we also impose a regularization term to the target variables. Defining the empirical label distribution of target variables as $f_k = P(\mathbf{y} = k) = \frac{1}{N}\sum_i q_{ik}$, we are able to enforce our preference for having balanced assignments by adding $KL(\mathbf{f}\|\mathbf{u})$ to the loss function, which conisders the uniform prior $\mathbf{u}$ for $\mathbf{f}$.

$$\mathscr{L} = KL(\mathbf{Q}\|\mathbf{P}) + KL(\mathbf{f}\|\mathbf{u}) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} q_{ik}\log\frac{q_{ik}}{p_{ik}} + q_{ik}\log\frac{f_k}{u_k},$$

Optimization: An alternating (EM-like) learning approach is utilized to optimize the objective function. In the expectation step, we derive an approximate closed form solution for estimating the target variables $\mathbf{Q}$. In the maximization step, the objective function is reduced to the standard cross entropy loss function to update the network parameters.

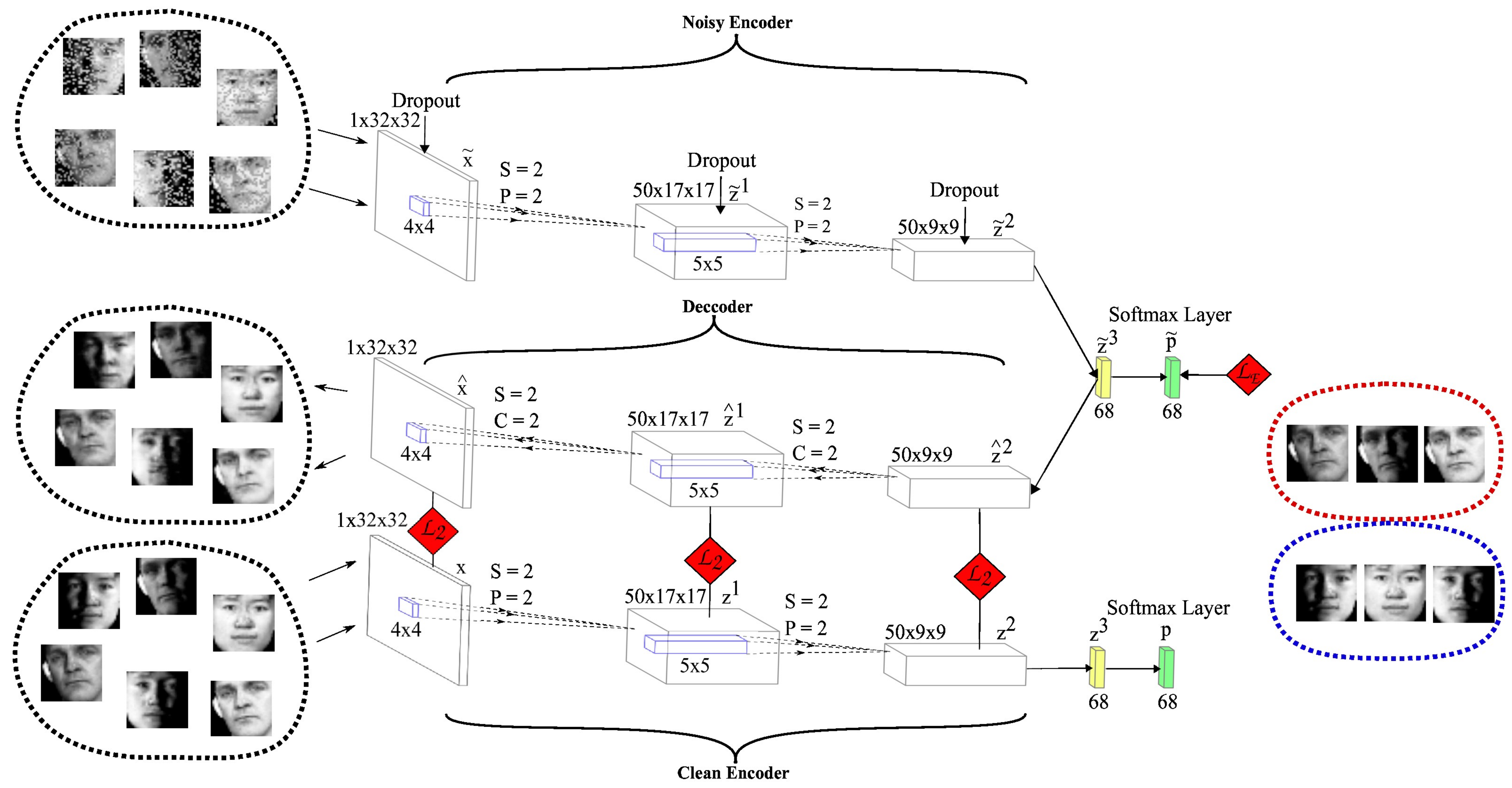
Algorithm 1: *DEPICT* Algorithm

1 Initialize $\mathbf{Q}$ using a clustering algorithm
2 **while** *not converged* **do**
3 $\quad \min_{\psi} \; -\frac{1}{N}\sum_{ik} q_{ik}\log\tilde{p}_{ik} + \frac{1}{N}\sum_{il}\frac{1}{|\mathbf{z}_i^l|}\|\mathbf{z}_i^l - \hat{\mathbf{z}}_i^l\|_2^2$
4 $\quad p_{ik}^{(t)} \propto exp(\boldsymbol{\theta}_k^T \mathbf{z}_i^L)$
5 $\quad q_{ik}^{(t)} \propto p_{ik}/(\sum_{i'} p_{i'k})^{\frac{1}{2}}$
6 **end**

Autoencoder Embedding: Deep embedding functions are useful for capturing the non-linear nature of input data. However, they may overfit to spurious data correlations and get stuck in an undesirable local minima. To avoid this overfitting problem, we design an autoencoder structures for clustering task, which has the reconstruction loss functions between every encoder and decoder layers as data-dependent regularizations.

Moreover, we compute target variables $\mathbf{Q}$ using the clean pathway, and model prediction $\tilde{\mathbf{P}}$ via the corrupted pathway (See *DEPICT* architecture). Hence, the clustering loss function $KL(\mathbf{Q}\|\tilde{\mathbf{P}})$ forces the model to have invariant features with respect to noise. In other words, the model is assumed to have a dual role: a clean model, which is used to compute the more accurate target variables; and a noisy model, which is trained to achieve noise-invariant predictions.

Architecture



Experimental Results

Dataset	*MNIST-full*		*MNIST-test*		*USPS*		*FRGC*		*YTF*		*CMU-PIE*		#
	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	HP
K-means	0.500*	0.534*	0.501*	0.547*	0.450*	0.460*	0.287*	0.243*	0.776*	0.601*	0.432*	0.223*	0
N-Cuts	0.411	0.327	0.753	0.304	0.675	0.314	0.285	0.235	0.742	0.536	0.411	0.155	0
SC-ST	0.416	0.311	0.756	0.454	0.726	0.308	0.431	0.358	0.620	0.290	0.581	0.293	0
SC-LS	0.706	0.714	0.756	0.740	0.681	0.659	0.550	0.407	0.759	0.544	0.788	0.549	0
AC-GDL	0.017	0.113	0.844	0.933	0.824	0.867	0.351	0.266	0.622	0.430	0.934	0.842	1
AC-PIC	0.017	0.115	0.853	0.920	0.840	0.855	0.415	0.320	0.697	0.472	0.902	0.797	0
SEC	0.779*	0.804*	0.790*	0.815*	0.511*	0.544*	-	-	-	-	-	-	1
LDMGI	0.802*	0.842*	0.811*	0.847*	0.563*	0.580*	-	-	-	-	-	-	1
NMF-D	0.152*	0.175*	0.241*	0.250*	0.287*	0.382*	0.259*	0.274*	0.562*	0.536*	0.920*	0.810*	0
TSC-D	0.651	0.692	-	-	-	-	-	-	-	-	-	-	2
DEC	0.816*	0.844*	0.827*	0.859*	0.586*	0.619*	0.505*	0.378*	0.446*	0.371*	0.924*	0.801*	1
JULE-SF	0.906	0.959	0.876	0.940	0.858	0.922	0.566	0.461	**0.848**	**0.684**	0.984	0.980	3
JULE-RC	0.913	0.964	**0.915**	0.961	0.913	0.950	0.574	0.461	**0.848**	**0.684**	**1.00**	**1.00**	3
DEPICT	**0.917**	**0.965**	**0.915**	**0.963**	**0.927**	**0.964**	**0.610**	**0.470**	0.802	0.621	0.974	0.883	0

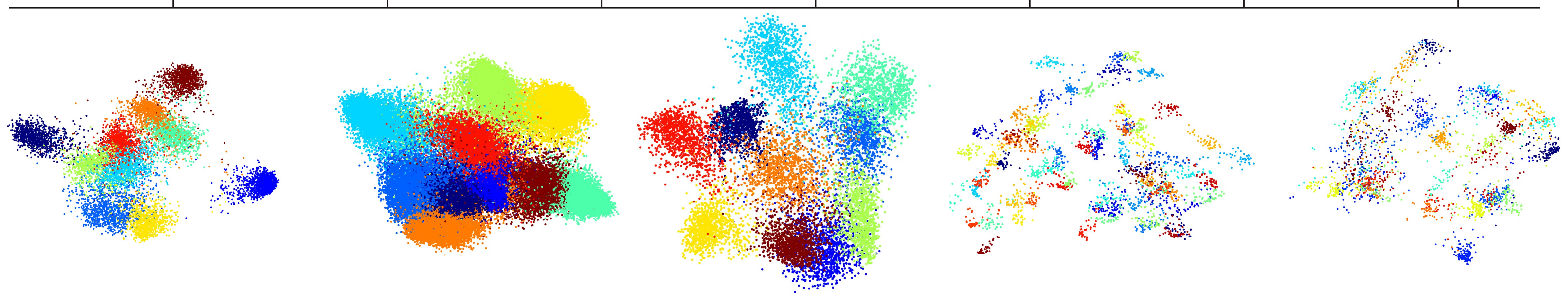


Figure 1: Visualization of *DEPICT* embedding subspace for *MNIST-full*, *MNIST-test*, *USPS* and *CMU-PIE* datasets.