Unsupervised deep embedding for clustering analysis (DEC)

(F) Unsupervised deep embedding for clustering Analysis.

Problem : n points $\{x_i \in X\}_{i=1}^{n}$ ⇒ k clusters.

cluster centroids : $\{\mu_j\}$; $j = 1 \dots k$, $\mu_j \in Z$

data space ↔ embedding (low dim)

Non linear mapping $f_\theta : X \to Z$

Solution : clustering with KL divergence :

① mapping & initialisation μ_j f_θ

Soft Assignment :

$$q_{ij} = \frac{\left(1 + \|z_i - \mu_j\|^2/\alpha\right)^{-\left(\frac{\alpha+1}{2}\right)}}{\sum_{j'} \left(1 + \|z_i - \mu_{j'}\|^2/\alpha\right)^{-\left(\frac{\alpha+1}{2}\right)}}$$

($z_i = f_\theta(x_i)$; α : inverse Hyperparam = 1)

KL Divergence Minimization :-

choice option for P ??

Loss, $L = KL(P\|Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$

$$\Rightarrow p_{ij} = \frac{q_{ij}^2/f_j}{\sum_{j'} q_{ij'}^2/f_{j'}} \quad // \quad f_j = \sum_i q_{ij}$$

(b) DEC cont.

Optimization:

$$\frac{\partial L}{\partial z_i} = \frac{\alpha+1}{\alpha} \sum_j \left(1 + \frac{\|z_i - \mu_j\|^2}{\alpha}\right)^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j)$$

$$\frac{\partial L}{\partial \mu_j} = -\frac{\alpha+1}{\alpha} \sum_i \left(1 + \frac{\|z_i - \mu_j\|^2}{\alpha}\right)^{-1} \times (p_{ij} - q_{ij})(z_i - \mu_j)$$

Accuracy Metric (classification setting)

$$ACC = \max_m \frac{\sum_{i=1}^{n} \mathbb{1}\{l_i = m(c_i)\}}{n}$$ // Latent class

$l_i \Rightarrow$ true label

$m \Rightarrow$ mapping of cluster c_i to label l_i

DINO: distillation with no labels

① DINO : Distillation with No Labels.

Approach: SSL with KD:

Student Network g_{θ_s} ; teacher Net g_{θ_t} → K-dimentional Output.

Softmax conversion of student Output.

$$P_s(x)^i = \frac{\exp\{g_{\theta_s}(x)^i / \tau_s\}}{\sum_{i'=1}^{K} \exp\{g_{\theta_s}(x)^{i'} / \tau_s\}} \quad \text{// student}$$

$$P_t(x)^i = \frac{\exp\{g_{\theta_t}(x)^i / \tau_t\}}{\sum_{j=1}^{K} \exp\{g_{\theta_t}(x)^j / \tau_t\}} \quad \text{// teacher.}$$

Optimization goal: $\min_{\theta_s} H(P_t(x), P_s(x))$ // $H(a,b) = -a \log b$

Actually: Local-to-Global view

$$\min_{\theta_s} \sum_{x \in \{x_1^g, x_2^g\}} \sum_{\substack{x' \in V \\ x' \neq x}} H(P_t(x), P_s(x'))$$

V → Set of images with different view of x.

$\{ x_1^g, x_2^g, x_1^L, \ldots x_n^L \}$

x_1^g, x_2^g ↓ global views { passed through teacher only }

$x_1^L \ldots x_n^L$ ↓ Local views { smaller dimentions } { student network }

Teacher Network Update:

$$\theta_t \leftarrow \lambda \theta_{t-1} + (1-\lambda)\theta_s \quad \text{// Polyak - Ruppert Avg}$$

Interesting Update.

Avoid collapse:

centering: $g_t(x) \leftarrow g_t(x) + c$ ← centering

$$c \leftarrow mc + (1-m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x)$$

Exp. Avg

⑰ Divide MIX

Divide Mix

Training Data $D = (X, Y) = \{(x_i, y_i)\}_{i=1}^{N}$

y_i → one hot.

Unlabed U

Cross entropy loss; $L(\theta) = \{L_i\}_{i=1}^{N} = \left\{ \sum_{c=1}^{C} y_i^c \log \left(p_{model}^c(x_i; \theta) \right) \right\}_{i=1}^{N}$

$p_{model}^c(x_i;\theta)$ → output for c

Penalize the entropy term:

$$fl = - \sum_{c} p_{model}^c(x;\theta) \log \left(p_{model}^c(x;\theta) \right)$$

// Avoid Near 0 noremalized loss.

minibatch labeled data / unlabeled data

$\{(x_b, y_b, w_b); b \in 1, \dots B\}$

$\{u_b; b \in (1, \dots B)\}$

gaussian component (smaller loss)

Data clean probability $w_i = p(g \mid L_i)$

loss value.

↓

compared with threshold τ

co-refinement: $\bar{y}_b = w_b y_b + (1 - w_b) p_b$

- y_b: provided label
- p_b: clean probability; Avg across different Augmentation (prediction by net)

Applying Sharpening:

$$\hat{y}_b = \text{sharpen}(\bar{y}_b, T) = \frac{\bar{y}_b^{c\,\frac{1}{T}}}{\sum_{c=1}^{C} \bar{y}_b^{c\,\frac{1}{T}}} \text{ for all } \forall c$$

Pair of sample (x_1, x_2) corresponding label (p_1, p_2)

mixed (x', p')

$$\lambda \sim \text{Beta}(\alpha, \alpha)$$
$$\lambda' = \max(\lambda, 1-\lambda)$$
$$x' = \lambda' x_1 + (1-\lambda') x_2$$
$$p' = \lambda' p_1 + (1-\lambda') p_2$$

Kind of Augmentation. (x' closer to x_1 than x_2)

Loss function:

{ Cross entropy loss } $\mathcal{L}_x = \frac{1}{|X'|} \sum_{x,p \in X'} \sum_c p_c \log\left(p_{model}^c(x;\theta)\right)$

{ consistency across net } $\mathcal{L}_u = \frac{1}{|U'|} \sum_{x,p \in U'} \| p - p_{model}(x,\theta) \|_2^2$

{ Regularization } $\mathcal{L}_{reg} = \sum_c \pi_c \log\left(\pi_c \Big/ \frac{1}{|X'|+|U'|} \sum_{x \in X'+U'} p_{model}^c(x,\theta)\right)$

(π_c: $1/c$ uniform)

MixMatch ⓟ

MixMatch

Consistency Regularization

$$\| p_{model}(y \mid \text{Augment}(x); \theta) - p_{model}(y \mid \text{Augment}(x); \theta) \|_2^2$$

Notation

Labeled data X ; unlabeled data $\mathcal{U}$

$$X', \mathcal{U}' = \text{MixMatch}(X, \mathcal{U}, T, K, \alpha)$$

$$\mathcal{L}_x = \frac{1}{|X'|} \sum_{x \in X'} H(p, p_{model}(y \mid x; \theta))$$

$$\mathcal{L}_u = \frac{1}{L|\mathcal{U}'|} \sum_{u,q \in \mathcal{U}'} \| q - p_{model}(y \mid u; \theta) \|_2^2$$

Augmentation: $\begin{cases} \hat{x}_b = \text{Augment}(x_b) \\ \hat{u}_{b,k} = \text{Augment}(u_b) \; ; \; k \in 1 \dots K \end{cases}$

Label guessing: $\bar{q}_b = \frac{1}{K} \sum_{k=1}^{K} p_{model}(y \mid \hat{u}_{b,k}; \theta)$

Sharpening $(p, T) := p_i^{1/T} \Big/ \sum_j p_j^{1/T}$

MixUp: Same as DivideMix.

Temporal Cycle-consistency learning

Ⓟ Temporal cycle-consistency learning

Notation: frame Sequence $S = \{s_1, s_2 \cdots s_N\}$ (frame)

Embedding $u_i = \phi(s_i; \theta)$ | image based??

Two video sequences: S, T (N, M length.)

Embedding $\{u_1, u_2 \cdots u_N\}$, $\{v_1, v_2 \cdots v_M\}$

U V

Cycle Consistency: $u_i \in U$

find $v_j = \arg\min_{v \in V} \|u_i - v\|$ // Nearest neighbor of u_i

Repeating $u_k = \arg\min_{u \in U} \|v_j - u\|$ // Nearest Neighbor of v_j

The points are cycle Consistent iff $i \equiv k$

track it to U set.

Cyck back classification:

Soft Nearest Neighbor: $\tilde{v} = \sum_{j=1}^{M} \alpha_j v_j$ (input point)

Softmax weight: $\alpha_j = \frac{e^{-\|u_i - v_j\|^2}}{\sum_{k}^{M} e^{-\|u_i - v_k\|^2}}$

N class classification problem: Logit $x_k = -\|\tilde{v} - u_k\|_2^2$

$\hat{y} = \text{softmax}(x)_k$

{ the smaller the better

Cross - Entropy loss:

only the $u_i \uparrow$ is ↑

$$L_{cbc} = -\sum_{j}^{N} y_i \log(\hat{y}_j)$$

Cycle back - regression:

$\in \cup$

Maximity Similarity vectors $\beta_k = \dfrac{\|\tilde{v} - u_k\|^2}{\sum_{j=1}^{K} \|\tilde{v} - u_k\|^2}$

variance

regularization

peaky around i

Regularizer

$$L_{Cbrc} = \frac{|i - \mu|^2}{\sigma^2} + \lambda \log \sigma$$

where,
$$\mu = \sum_{k=1}^{N} \beta_k k$$
$$\sigma^2 = \sum_{k=1}^{N} \beta_k * (k - \mu)^2$$

The losses are differentiable

Backpropagation

⑥ The whitening loss

The whitening loss

Embedding $z = f(x, \theta)$

Goal, $\begin{cases} \min_{\theta} E\left[dist(z_i, z_j)\right] \\ \text{S.t.} \quad cov(z_i, z_i) = cov(z_j, z_j) = I \end{cases}$

$$dist(z_i, z_j) = \left\| \frac{z_j}{\|z_j\|_2} - \frac{z_i}{\|z_i\|_2} \right\|_2^2$$

$$= 2 - 2\frac{\langle z_i, z_j \rangle}{\|z_i\|_2 \|z_j\|_2}$$

Original Image No. N

Batch $B = \{x_1, x_2 \cdots x_K\}$

$K = Nd$ → No of positives.

$V = \{v_1, v_2 \cdots v_K\}$

Proposed W-MSE:

$$L_{W\text{-}MSE}(V) = \frac{2}{Nd(d-1)} \sum dist(z_i, z_j);\ (i,j) \in \text{Positive}$$

whitening (v)

$$\text{whitening}(v) = W_V(v - \mu_v)$$

$$\mu_v = \frac{1}{K}\sum_K v_k$$

$$\Sigma_v = \frac{1}{k-1}\sum_k (v_k - \mu_v)(v_k - \mu_v)^T$$

$$W_V^T W_V = \Sigma_V^{-1} \quad \text{// inverse of covariance Matrix.}$$

Memory bank Memory Bank

Parametric Classifier:

n images $x_1 \dots x_n$: n classes.

feature $v_j = f_\theta(x_i)$ // NN

$$P(i|v) = \frac{\exp(w_i^T v)}{\sum_{i=1}^{N} \exp(w_i^T v)}$$

w_i → weight vector for class i

Non Parametric classifier:

$$P(i|v) = \frac{\exp(v_i^T v)}{\sum_{j=1}^{N} \exp(v_i^T v)}$$

v_i → the instances itself .!?

Learning Objective:

$$J(\theta) = -\sum_{i=1}^{n} \log P\left(i \mid f_\theta(x_i)\right) \quad \text{// minimize}$$

Augment version to itself ?!

Equivalent to: $\prod_{i=1}^{n} P_\theta\left(i \mid f(x_i)\right)$ // maximization

NCE: $$P(i|v) = \frac{\exp(v^T f_i / \tau)}{Z_i}$$

$$Z_i = \sum_{j=1}^{N} \exp(v_j^T f_i / \tau)$$

$$h(i,v) := P(D=1|i,v) = \frac{P(i|v)}{P(i|v) + m P_n(i)}; P_n = \frac{1}{n}$$

m times more noise sample

$$J_{NCE}(\theta) = -E_{P_d}[\log h(i,v)] - m E_{P_n}[\log(1 - h(i,v'))]$$

→ other image feature

Added for regularize

$$z := z_i = n E_j[\exp(v_j^T f_i / \tau)] = \frac{n}{m} \sum_{k=1}^{m} \exp(v_j^T f_i / \tau)$$

Proximal Regularization:

Loss function: $-\log h(i, v_i^{t-1}) + \lambda \| v_i^{(t)} - v_i^{(t-1)} \|_2^2$

⑤ PIRL

PIRL (pretext invariant representation learning)

Notation: Dataset $D = \{I_1, I_2 \cdots I_D\}$

$I_n \in \mathbb{R}^{W \times H \times 3}$

Transformation set, T

Network $\phi_\theta(\cdot)$

$V_I = \phi_\theta(I)$

Target optimization:

$$L_{inv}(\theta, D) = E_{t \sim P(T)} \left[\frac{1}{|D|} \sum_{I \in D} L(V_I, V_{I_t}) \right]$$

contrastive target:

$$L_{co}(\theta, D) = E_{t \sim P(T)} \left[\frac{1}{|D|} \sum_{I \in D} L_{co}(V_I, z(t)) \right]$$

[measure properties of transformation + Keep semantic irrelevant info.]

loss function:

$$h(V_I, V_{I_t}) = \frac{\exp\left(\frac{s(V_I, V_{I_t})}{\tau}\right)}{\exp\left(\frac{s(V_I, V_{I_t})}{\tau}\right) + \sum_{I' \in D_N} \exp\left(\frac{s(V_{I_t}, V_{I'})}{\tau}\right)}$$

$$L_{NCE}(I, I^t) = -\log\left[h(f(V_I), g(V_{I_t}))\right] - \sum_{I' \in D_N} \log\left[1 - h(g(V_{I'}), f(V_I))\right]$$

final loss function:

step 2.

$$L(I, I^t) = \lambda L_{CNF}(m_I, g(V_{I^t})) + (1-\lambda) L_{ENCE}(m_I, f(V_I))$$

step 1 (i)

kinda two step

→ (i) $f(V_I)$ similar to m_I // dampening.

→ (ii) $f(V_I)$, $f(V_{I'})$ dissimilar.

ⓟ EqCo

EqCo (Equivalent Contrastive)

$$\mathcal{L}_{INCE} = \mathop{\mathbb{E}}_{\substack{q \sim D,\ k_i \sim \bar{D} \\ k_0 \sim D(q)}} \left[-\log \frac{e^{(q^T k_0 - m)/\tau}}{e^{(q^T k_0 - m)/\tau} + \sum_{i=1}^{N} e^{q^T k_i / \tau}} \right]$$

(margin, temperature)

$q \rightarrow$ Query Sample Representation.

$k \rightarrow$ keys. $0 \rightarrow$ positive; $k_i \rightarrow$ Negative.

$D \rightarrow$ Data Distribution.

Equivalent rule: $m = \tau \log \frac{N}{K}$ (A constant !!; No of Negatives.; Temperature.; margin)

EqCo: **Batch Size**: Number of queries in 'N' per batch.

Negative Samples: K per query

Query Embedding 'q'

Key Embedding $x = \{x_i\}_{i=0,\dots K}$

Conditional $P(x_i | q)$

independent dist $P(x_i)$

$$P_{H_i}(x) = P(x_i|q) \prod_{j \neq i} P(x_j)$$

↳ H_i (k+1) candidate distribution for x.

$$\Pr\left[x \sim H_0 \mid q, x\right] = \frac{p^+ P_{H_0}(x)}{p^+ P_{H_0}(x) + p^- \sum_{i=1}^{K} P_{H_i}(x)}$$

$$= \frac{\frac{p^+}{p^-} \frac{p(k_0|q)}{p(k_0)}}{\frac{p^+}{p^-} \frac{p(k_0|q)}{p(k_0)} + \sum_{i=1}^{K} \frac{p(k_i|q)}{p(k_i)}}$$

Let, $\frac{p(k_i|q)}{p(k_i)} \propto e^{q^T k_i / \tau} \quad (i = 0, \cdots K)$

$$\frac{p^+}{p^-} = e^{-m/\tau}$$

then, $\mathcal{L}_{opt} \triangleq \mathbb{E}_{q,x} - \log \Pr\left[x \sim H_0 \mid q, x\right]$

$$\geq \log\left[1 + K e^{m/\tau}\right] - I(k_0, q)$$

$$I(k_0, q) \geq f_{bound}(m, K)$$

$$\triangleq \log\left(1 + K e^{m/\tau}\right) - \mathcal{L}_{opt}$$

$$\approx \log\left(1 + K e^{m/\tau}\right) - \mathop{\mathbb{E}}_{q \sim D, k_0 \sim D(q)} \log\left(1 + K e^{m/\tau} \frac{p(k_0)}{p(k_0|q)}\right)$$

⑯ Pseudo 3D CNN

Residual units: $x_{t+1} = h(x_t) + F(x_t)$

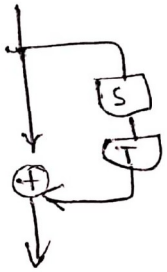
- x_{t+1} → output of t^{th} block.
- x_t → output of t
- $h(x_t)$ → identity map

P3D Block design:

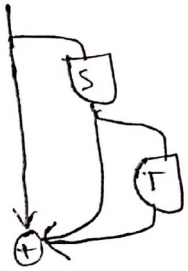
S → 2D spatial filter

T → 1d temporal filter.

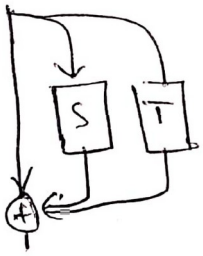
i) P3D - A



ii) P3D C



iii) P3D - B



① Partially view alignment RL with noise robust

Partially View Alignment RL with Noise robust

$$\{x_{i=1}^{i}\}_{i=1}^{v} = \{x_1^i, x_2^i, \dots x_N^i\}_{i=1}^{v}$$

Partially views aligned data.

$$\{X_i^i\}_{i=1}^{v} = \{x_i^i, v_i^i\}_{i=}^{v}$$

Align unaligned

Target Align : U utilizing A

Loss function:

$$L = \frac{1}{2r} \sum_{i=1}^{N} \{ p\, l_i^{pos} + (1-p)\, l_i^{neg} \}.$$

$$l_i^{pos} = d(a_i, a_i^2) = \| f_1(a_i^1) - f_2(a_i^2) \|$$

$$l_i^{neg} = \frac{1}{m} \max \left(m\, d^{v_2}(a_i^1, a_j^2) - d^{3/2}(a_i^1, a_j^2), 0 \right)$$

$$m = \frac{i}{N_p} \sum d(a_i^1, a_i^2) + \frac{1}{N_n} \sum d(a_i^1, a_j^2)$$

modified

Originally, $l_i^{ctr} = \max \left(m - d(a_i^1, a_j^2), 0 \right)^{2}$