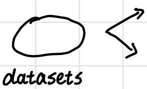
Hierarchical Dynamic Labeling (HDL)

Semi-supervised learning

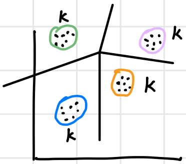


labeled → □ ← unlabeled

pesudo-label ←

□ : { predict network (classifier)
or
Represent network (feature extractor) ✓ Generalization ability ↑ }

k-NN label clusterability (unsupervised learning)



embedding space

Similar embedding → assemable

[metric learning (clip, Dino)]

Defined :

labeled dataset: $D=\{(x_n, y_n)\}_{n\in[N]}$ $[N]=\{1,2,\cdots,N\}$ $y_n\in\{1,2,\cdots,C\}$

unlabeled dataset: $D'=\{x'_m\}_{m\in[M]}$ y'_m : potential label

$\begin{cases} \forall m\in[M], y'_m\in[C] \\ \exists m\in[M], y'_m\notin[C] \end{cases}$ $\xrightarrow{\text{closed-set}}$ $\forall m\in[M], y'_m\in\{1,2,\cdots,C\}$

Define 1 : label clusterability (k, δ_k)

Dataset D 、 image encoder $f(\cdot)$

→ embeddings $\mathcal{X}=f(D)$

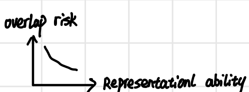
$\forall x\in\mathcal{X}$, $P(x$ and k-nearest neighbors → Same class$)\geq 1-\delta_k$

↘ $D \to (k, \delta_k)$ δ_k: D Violate clusterability probability

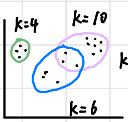
(1) $\delta_k=0$, $D \to$ k-NN label clusterability

(2) $k\uparrow \to \delta_k\uparrow$

(3) Representational ability ↓ → $\delta_k\uparrow$

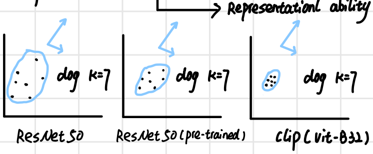


(2)



$k\uparrow$ Risk ↑ $\delta_k\uparrow$

(3)



Iterative algorithm process :

① Ideal label clusterability (paper 4.1 (KNN-DU))

Ideal : Strict clustering → lables

labeled embeddings $D=\{(x_n, y_n)\}_{n\in[N]}$

unlabeled embeddings $D'=\{x'_m\}_{m\in[M]}$

Since ideal So $D \to$ k-NN label clustability

$x'_m, m\in[M] \longrightarrow y'_m, m\in[M]$

x'_m k_1 NN from D = $x'_{m1}, x'_{m2}\cdots x'_{mk_1}$

Corresponding labels: $y'_{m1}, y'_{m2}\cdots y'_{mk_1}$

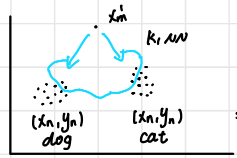
label convert : integer ⟶ one-hot

Vote : $y'_m = \underset{i\in[C]}{\arg\max}\left(\frac{1}{k_1}\sum_{j}^{k_1} y'_{mj}\right)[i]$, $y'_{mj}\in R^C$

Traversal $\begin{cases} 1: i\in[C] & C: \text{number of labels} \\ 2: \sum_j^{k_1} y'_{mj} : & k_1 \text{ nearest neighbors labels from } D \text{ one-hot form addition} \end{cases}$

Geometric perspective :

classes	integer	one-hot
dog	0	10
cat	1	01



$([1,0]\times 2+[0,1]\times 3)/5$

$\Rightarrow \arg\max([\frac{2}{5}, \frac{3}{5}])$

$\Rightarrow 1 \Rightarrow$ cat $= y'_m$

Weakness :

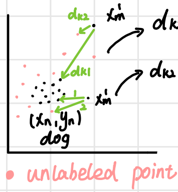
1: Rely on highly clustering !

2: Step by step import unlabeled point

Ignore the relationship between them !

$y_n\in[C]$, $y'_m\in[C]$ Same of label space

⇒ D mix D' follow clustrability in embedding space



• unlabeled point

$d_{k2} \ll d_{k1}$ Consider • is essential

$d_{k2} > d_{k1}$ Even so, • is essential

⇒ k constant, import •

• needs less

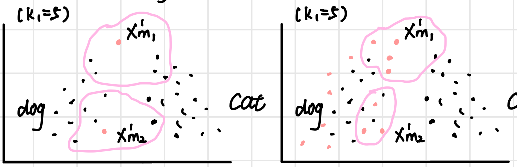
We can proved clustrability be allowed ↓ Since • needs less

⟶ Continous ...

we assume x'_m authentic label is dog

$\Rightarrow$ how ensure x'_m labeling correctly when clustrability $\downarrow$

Simultaneous analysis of near and far points x'_{m_1}, x'_{m_2}



• unlabeled point (But no others)

• unlabeled point (have others help)

$\Rightarrow$ I think • as fault-tolerant when clustrability $\downarrow$

② unlabeled embedding help each others (paper 4.2 (KNN))

May $x'_{m_1}, x'_{m_2} \cdots x'_{mk}$ is unlabeled embedding

$\exists j \in [k] \to x'_{mj} \in D' \to$ how we know its label

$\Rightarrow$ Depth-First Search (DFS) for tree structure

$x'_m \; y'_m (?)$ $\quad y'_m = \underset{i \in [C]}{\operatorname{argmax}} \left(\frac{1}{k}\sum_j^k y'_{mj}\right)[i]$

$x'_{m1} \; y'_{m1} \quad \cdots \quad x'_{mi} \; y'_{mi} (?) \quad \cdots \quad x'_{mk} \; y'_{mk}$ $\quad y'_{mi} = \underset{j \in [C]}{\operatorname{argmax}} \left(\frac{1}{k}\sum_a^k y'_{mia}\right)[j]$

$x'_{mi1} \; y'_{mi1} \quad \cdots \quad x'_{mia} \; y'_{mia} (?) \quad \cdots \quad x'_{mik} \; y'_{mik}$

Exhaustively $\to$ find all labels
or
proportional threshold $\to$ partial labels compute manager

HDL : Dynamically increase determined labels

weakness : fall into
- loop search
- Multi branch repeated search

③ Hierarchical Dynamic labeling (paper 4.3 HDL)

Try best to avoid fall : (core)

(1) Prioritize labeling clearer ingredients (P(fall) minimum)

(2) Dynamically update D ($D' \downarrow$ $D \uparrow$) $\Rightarrow P$(the rest, fall) $\downarrow$

(1) self optimal

$D = \{(x_n, y_n)\}_{n \in [N]}$ $\quad D' = \{x'_m\}_{m \in [M]}$

$x'_m \; y'_m (?)$ $\quad$ x'_m subset clarity L_i

✓ ? ✓ $\quad$ subset clearer $\uparrow$ fall $\downarrow$

D', KNN, belong to D, is $L_1, L_2, \cdots L_m$

clearest : $L_m = \max\{L_1, L_2, \cdots L_m\}, m \in [M]$

L_m { one / multi (s) } $\Rightarrow$ unlabeled embedding $x'_{max,1}, \cdots, x'_{max,s}$, $s \le M$

(2) overall optimal

$x'_{max,1}, \cdots, x'_{max,s}$ $s \le M$ (self-optimal)

arbitrary select ?

Assume labeling $x'_{max,i}$, $i \in [s] \to x'_{max,i}$ from D' to D

Then the rest of self-optimal's l may change

$[l_{i,1}, \cdots, l_{i,i-1}, l_{i,i+1}, \cdots l_{i,s}] \in R^{s-1}$ expand

$$l = \begin{bmatrix} l_{1,2} & \cdots & l_{1,i} & l_{1,i+1} & \cdots & l_{1,s} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ l_{i,1} & \cdots & l_{i,l_{i-1}} & l_{i,l_{i+1}} & \cdots & l_{i,s} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ l_{s,1} & \cdots & l_{s,l_{i-1}} & l_{s,l_i} & \cdots & l_{s,l_{s-1}} \end{bmatrix} \in R^{s \times (s-1)}$$

$l[i]$: labeling the i-th, the rest of $D' \to l$

$\underset{i \in [s]}{\operatorname{argmax}}\, \operatorname{sum}(l[i]) \to$ Rest-optimal

self-optimal + Rest-optimal = overall-optimal

④ Adaptive k (paper 4.4)

balance
- KNN : $k \uparrow$ clustrability $\downarrow$
- Majority voting : $k \uparrow$ robust $\uparrow$ Reliable $\uparrow$

Ref [43] (2022 ICML) :

"Detecting corrupted labels without training a model to predict"

$\longrightarrow$ Continous ...

self-optimal
overall-optimal

2022 ICML: (lower bound)

$p(\text{Vote is correct} \mid k) \geq (1-\delta_k) \cdot I_{1-e}(k+1-k', k'+1)$

$(1-\delta_k)$: Quality of features

Based on it

$\Rightarrow p(k) \geq \mu_k \cdot I_{1-e}(k+1-k', k'+1)$

μ_k: p(clustrability)

e: p(error labels in embedding set)

k': $[(k+1)/2]-1$

$I_{1-e}(k+1-k', k'+1)$. regularized incomplete beta function

$\mu_k = (1-\delta_k)$ = Quality of embedding

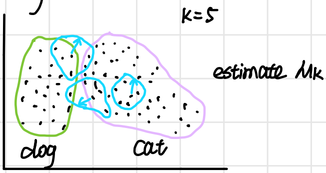
$$I_{1-e}(k+1-k', k'+1) = \frac{(k+1)!}{(k-k')!k'!} \int_0^{1-e} t^{k-k'}(1-t)^{k'} dt$$

$$k\uparrow \Rightarrow \begin{cases} I_{1-e}(k+1-k', k'+1) \uparrow \\ \text{clustrability} \downarrow \Rightarrow \mu_k \downarrow \end{cases}$$

Qutification μ_k:

From labeled embedding space

randomly select instances to statistics



(backbone: clip)