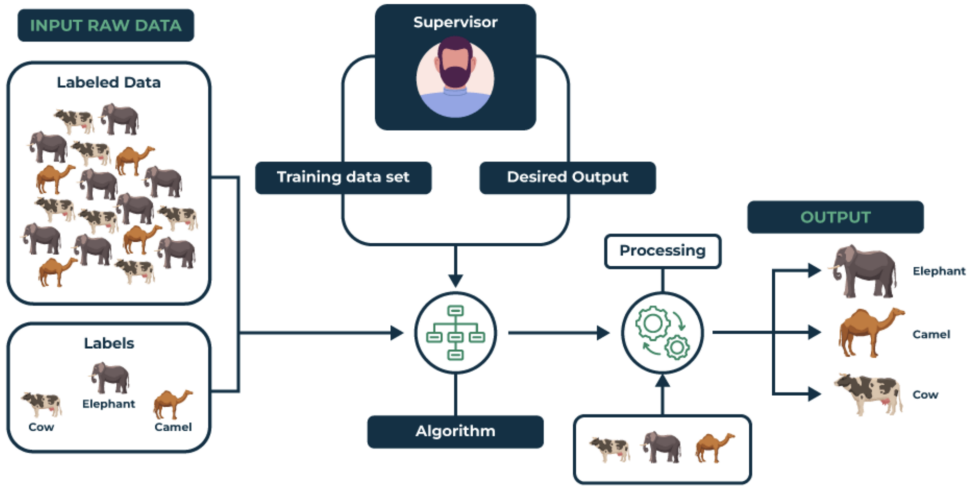
Contrastive Learning

Ziruo(Rosie) Zhao

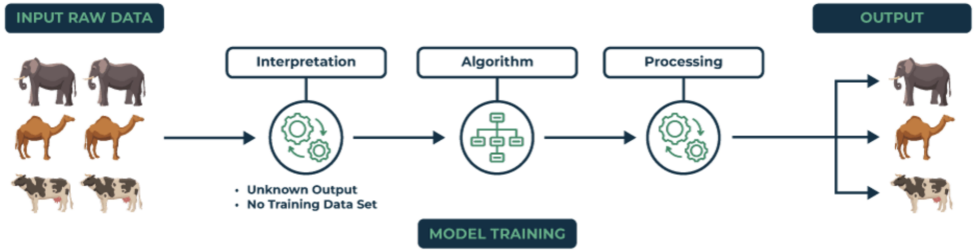
October 8 2024

Machine learning

1. Supervised learning
2. Unsupervised learning
3. Semi-supervised learning



(a) Supervised



(b) Unsupervised

Self-Supervised Learning

What is self-supervised learning?
Self-supervised learning (SSL) is a paradigm in machine learning where a model is trained on a task **using the data itself to generate supervisory signals**, rather than relying on external labels provided by humans.

Visual Representation Learning

Facial recognition:

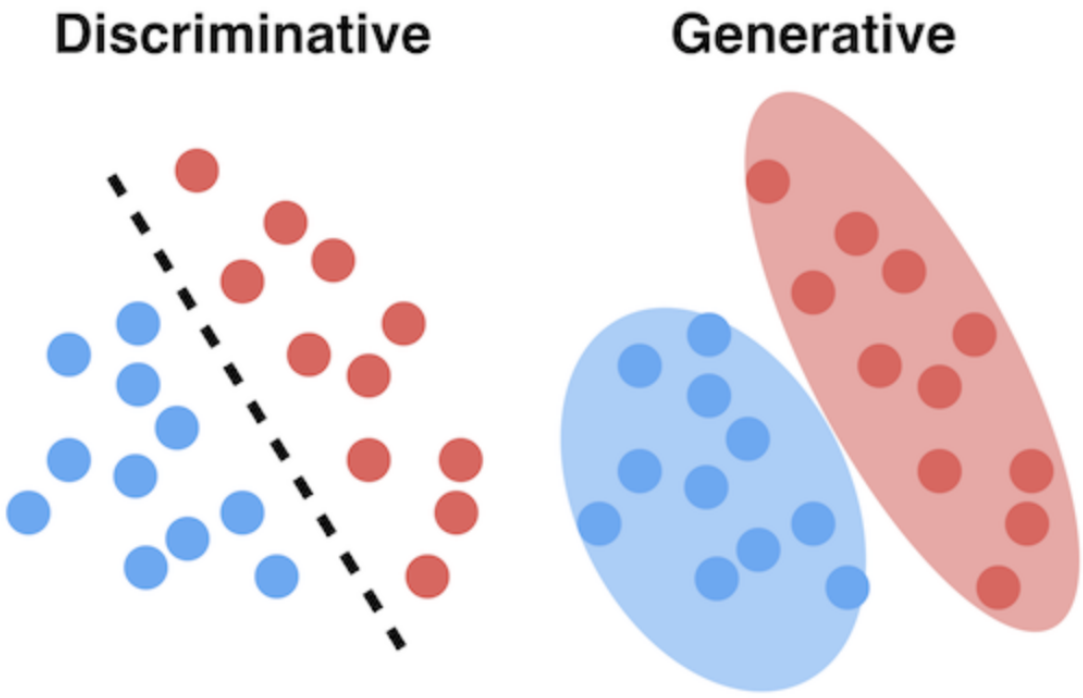
- Distance between the eyes
- Distance from the forehead to the chin
- Distance between the nose and mouth
- Depth of the eye sockets
- Shape of the cheekbones
- Contour of the lips, ears, and chin

Then convert the face recognition data into a string of numbers or points called a "faceprint". Each person has a unique "faceprint", similar to a fingerprint.

Unsupervised Visual Representation Learning

Two approaches:

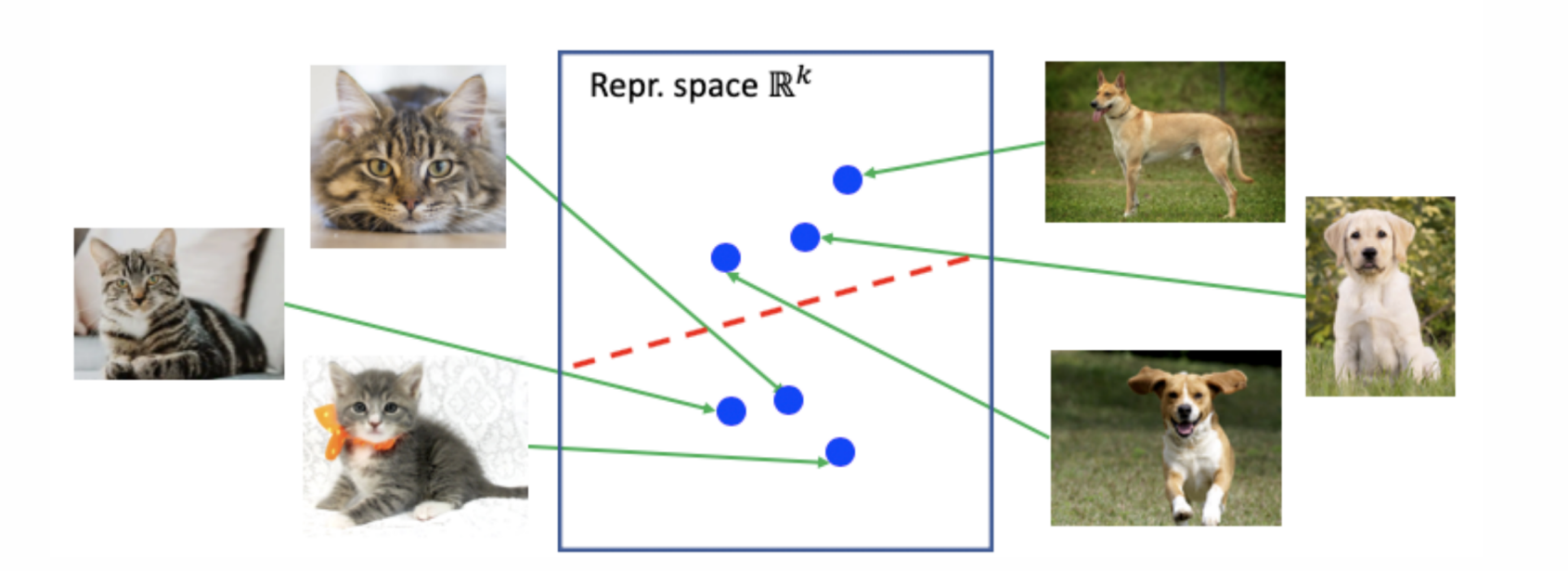
- Generative: model the distribution of individual classes
- Discriminative: learn the boundary between classes



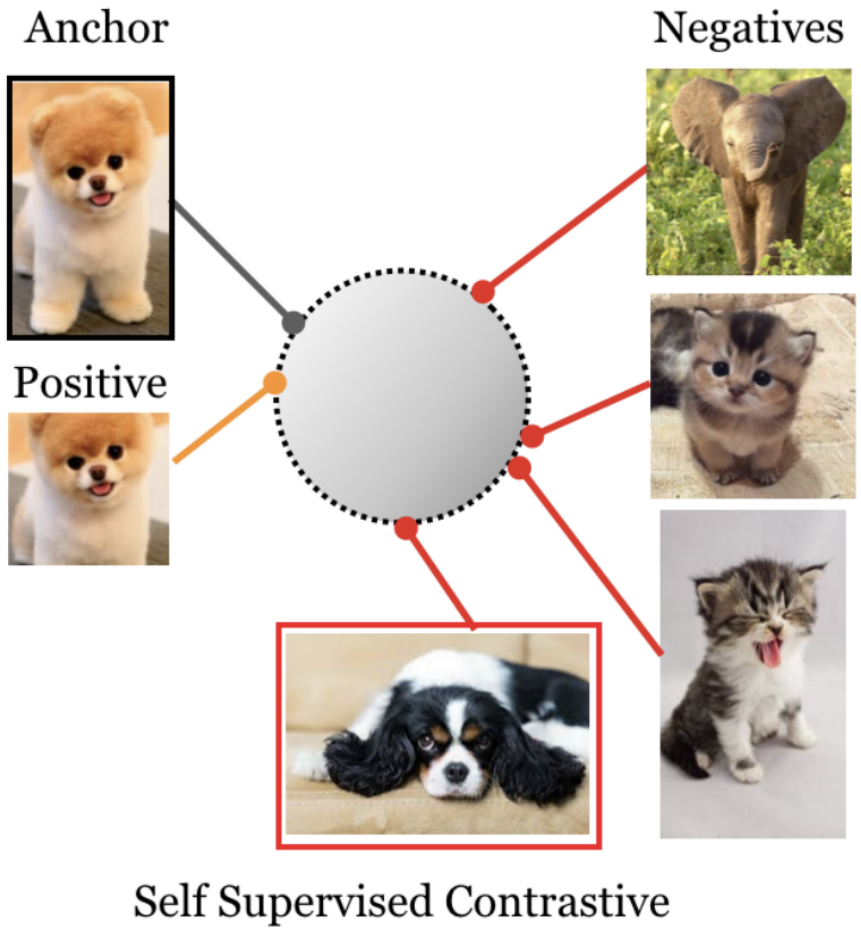
Contrastive learning

Learn feature extractors by

1. minimizing the distance between the representations of positive pairs, or samples that are similar in some sense,
2. maximizing the distance between representations of negative pairs, or samples that are different in some sense.



Contrastive learning: Sample Dataset



Data Augmentation

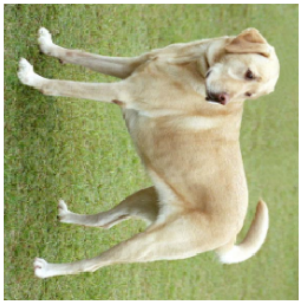
(a) Original

(b) Crop and resize

(c) Crop, resize (and flip)

(d) Color distort. (drop)

(e) Color distort. (jitter)



(f) Rotate $\{90°, 180°, 270°\}$

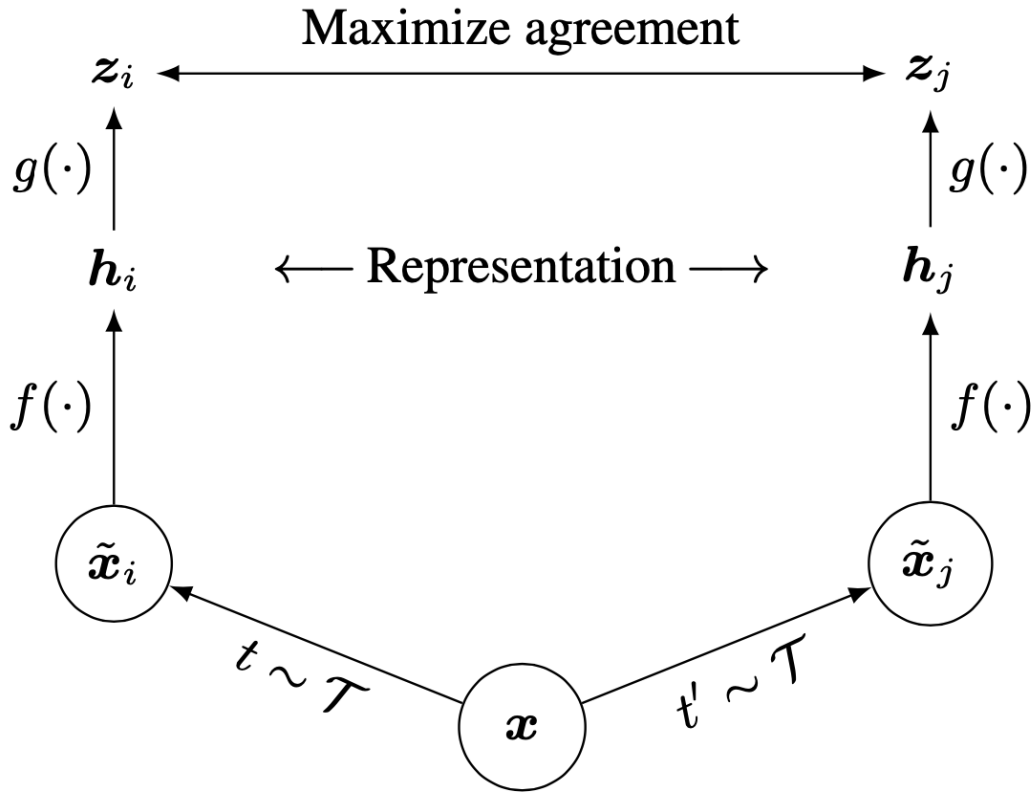
(g) Cutout

(h) Gaussian noise

(i) Gaussian blur

(j) Sobel filtering

A simple framework for contrastive learning



Similarity $sim(z_i, z_j) = \frac{z_i{}^T z_j}{||z_i|| \cdot ||z_j||}$

Contrastive Loss

The loss function for a positive pair of examples (i, j) is defined as

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

Algorithm

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of $f, g, \mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \ldots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i, j)$ **as** $\ell(i, j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

A Theoretical Framework of Contrastive Learning

Assumption

- $\mathcal{X}$ denote the set of all possible data points.
- $\mathcal{C} = \{c_1, c_2, ...\}$ denote the set of all latent classes.
- $D_c(x)$ is a probability distribution over $\mathcal{X}$ for class c, captures how relevant x is to class c.
- Assume a distribution ρ over the classes.

Training data

- Randomly pick some class c according to ρ.
- For each class c, i.i.d draw similar pair(x, x^+) from the distribution $D_c(x)$.
- $D_{sim}(x, x^+) = \mathbb{E}_\rho \; D_c(x) D_c(x^+)$.
- Negative samples ${x_1}^-, ..., {x_k}^-$ are i.i.d draws from the marginal of D_{sim}.
- $D_{neg}(x^-) = \mathbb{E}_\rho \; D_c(x^-)$.

Unsupervised Contrastive Loss

- Unsupervised Loss is
  $L_{un}(f) = \mathbb{E}[\mathcal{L}(\{f(x)^T(f(x^+) - f(x_i^-))\}_{i=1}^k)]$.
- Empirical unsupervised loss with M samples
  $(x_j, x_j^+, x_j 1^- ... x_j k^-)_{j=1}^M$ is
  $\hat{L}_{un}(f) = \frac{1}{M}\Sigma\mathcal{L}(\{f(x_j)^T(f(x_j^+) - f(x_{ji}^-))\}_{i=1}^k)$.

The algorithm finds a f from $\mathcal{F}$ that minimizes the empirical unsupervised loss: $\hat{f} \in arg\ min\ \hat{L}_{un}(f)$.

Downstream Task

Setup:

- Supervised task $\mathcal{T}$ consists of $k+1$ classes $\{c_1, ..., c_{k+1}\} \subseteq \mathcal{C}$.
- Labeled dataset: m samples i.i.d draws from following process
  1. Pick class c according to $D_{\mathcal{T}}$,
  2. Draw sample x from D_c,
  3. form a labeled pair (x, y).
- m samples $(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)$

Downstream Task

- Optimal representation function $\hat{f}$
- Consider linear classification: A multi-class classifer is $g : \mathcal{X} \to \mathbb{R}^{k+1}$
- Supervised loss of g is $L_{sup}(\mathcal{T}, g) = \mathbb{E}_{D_{\mathcal{T}}}[\mathcal{L}(\{g(x)_c - g(x)_{c'}\}_{c \neq c'})]$
- Train a matrix $W \in \mathbb{R}^{(k+1) \times d}$, $g(x) = Wf(x)$.
- The best W is chosen for f: $L_{sup}(\mathcal{T}, f) = inf_W \ L_{sup}(\mathcal{T}, Wf)$

Averaged Supervised Loss

- Mean Classifier W^{μ}: c^{th} row is the mean $\mu_c = \mathbb{E}_{D_c}[f(x)]$.
- For a function f, supervised loss of its mean classifier: $L^{\mu}_{sup}(\mathcal{T}, f) = L_{sup}(\mathcal{T}, W^{\mu} f)$
- Averaged supervised loss of f is $L_{sup}(f) = \mathbb{E}_{\{c_i\}}[L_{sup}(\{c_i\}_{i=1}^{k+1}, f) | c_i \neq c_j)]$
- Averaged supervised loss of its mean classifier is $L^{\mu}_{sup}(f) = \mathbb{E}_{\{c_i\}}[L^{\mu}_{sup}(\{c_i\}_{i=1}^{k+1}, f) | c_i \neq c_j)]$

Bound on the downstream performance

$$L_{sup}(\hat{f}) \leq \alpha L_{un}(f) + \eta G_M + \delta \quad \forall f \in \mathcal{F}$$

- α, η, δ are constants depending on the distribution of classes.
- $G_M \to 0$ as $M \to \infty$.

A Theoretical Framework with Spectral Contrastive Loss

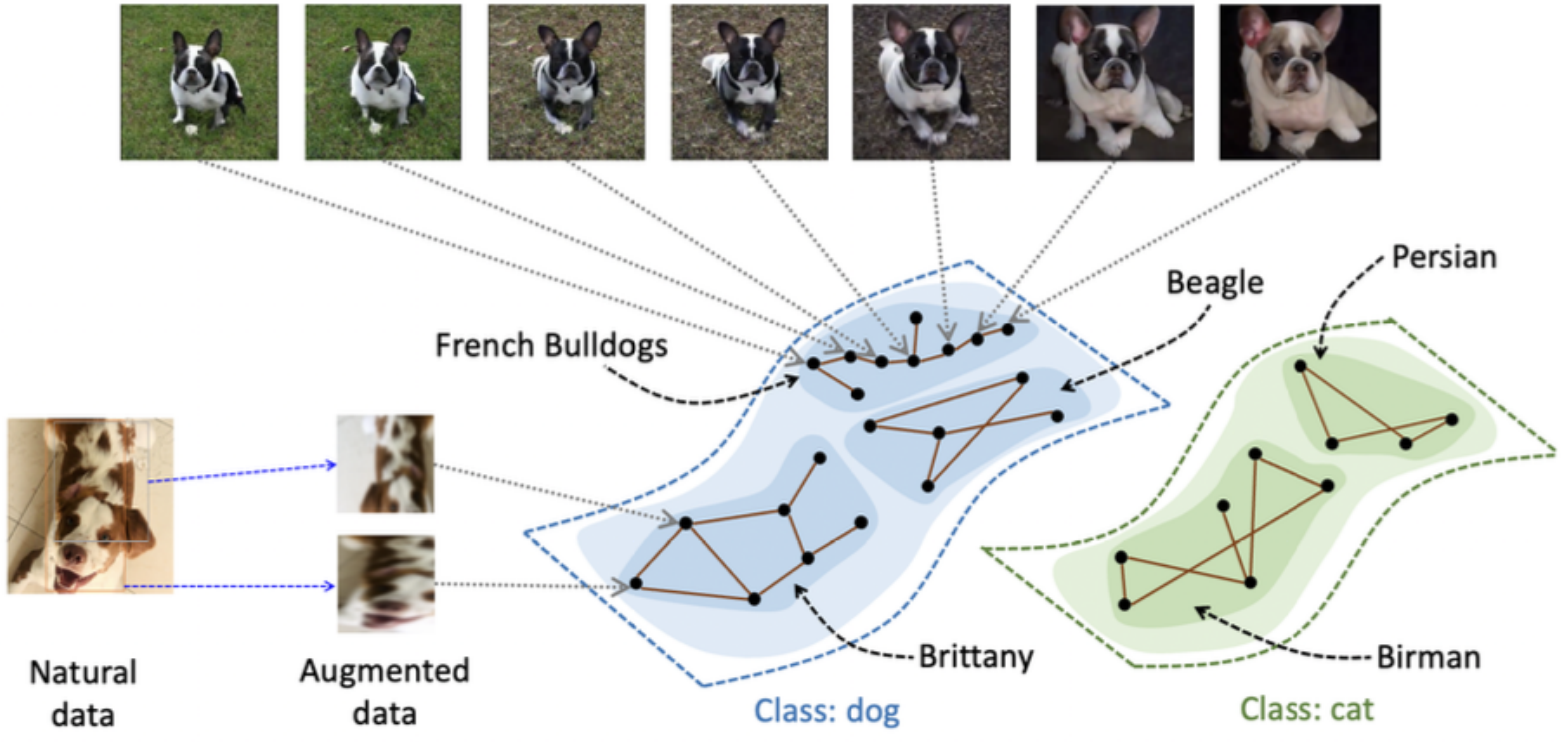
Assumption

Without assuming conditional independence of positive pairs, using augmentation graph on data.

- Natural data $\overline{\mathcal{X}} \sim \mathcal{P}_{\overline{\mathcal{X}}}$, finite but exponentially large
- r classes
- Ground-truth $y : \overline{\mathcal{X}} \to [r]$
- Distribution of augmentation $\mathcal{A}(\cdot|\overline{x})$
- Augmented data $\mathcal{X}$, $N = |\mathcal{X}|$

Population Augmentation Graph

- Population augmentation graph $G(\mathcal{X}, \omega)$
- Vertex set is all augmented data $\mathcal{X}$
- For any $x, x' \in \mathcal{X}$, edge weight $w_{xx'} = \mathbb{E}_{\mathcal{P}_{\overline{\mathcal{X}}}}[\mathcal{A}(x|\overline{x})\mathcal{A}(x'|\overline{x})]$, the probability that a random positive pair (x, x') from the same image.



Eigendecomposition

- Adjacency matrix $A \in \mathbb{R}^{N \times N}$, $A_{xx'} = w_{xx'}$
- $w_{xx'}$: the probability of a random positive pair being (x, x')
- Total weights of x: $w_x = \Sigma_{x'} w_{xx'}$
- w_x: the probability of a random augmented datapoint being x.
- Diagonal matrix $D \in \mathbb{R}^{N \times N}$, $D_{xx} = w_x$
- Normalized adjacency matrix $\overline{A} = D^{-1/2} A D^{-1/2}$
- Find eigenvectors of $\overline{A}$ corresponding to k largest eigenvalues.

Eigendecomposition

$$F = \begin{bmatrix} | & & | \\ v_1 & \dots & v_k \\ | & & | \end{bmatrix} = \begin{bmatrix} -u_1- \\ \dots \\ -u_n- \end{bmatrix} \leftarrow \begin{bmatrix} -f(x_1)- \\ \dots \\ -f(x_n)- \end{bmatrix}$$

- F is the matrix of first k eigenvectors
- Denote each row of F as u_x
- Reparameterize $u_x = w_x^{1/2} f(x)$
- Goal: minimizing "matrix factorization loss"

$$\min_{F \in \mathbb{R}^{N \times k}} \mathcal{L}_{\text{mf}}(F) := \left\| \overline{A} - FF^\top \right\|_F^2$$

Spectral Contrastive Loss

$$\begin{aligned}\mathcal{L}_{\mathrm{mf}}(F) &= \sum_{x,x' \in \mathcal{X}} \left(\frac{w_{xx'}}{\sqrt{w_x w_{x'}}} - u_x^\top u_{x'} \right)^2 \\ &= \sum_{x,x' \in \mathcal{X}} \left(\frac{w_{xx'}^2}{w_x w_{x'}} - 2 \cdot w_{xx'} \cdot f(x)^\top f(x') + w_x w_{x'} \cdot \left(f(x)^\top f(x') \right)^2 \right)\end{aligned}$$

$$\mathcal{L}_{\mathrm{mf}}(F) = \mathcal{L}(f) + \text{const}$$

$$\text{where } \mathcal{L}(f) \triangleq -2 \cdot \mathbb{E}_{x,x^+} \left[f(x)^\top f(x^+) \right] + \mathbb{E}_{x,x^-} \left[\left(f(x)^\top f(x^-) \right)^2 \right]$$

References

- **A Simple Framework for Contrastive Learning of Visual Representations.** Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Proceedings of the 37th International Conference on Machine Learning (ICML), 2020.
- **A Theoretical Analysis of Contrastive Unsupervised Representation Learning.** Pratyush Maini, Ekin Dogus Cubuk, Alexei A. Efros, Pieter Abbeel, and Ishan Misra. Advances in Neural Information Processing Systems (NeurIPS), 2022.
- **Provable Guarantees for Self-Supervised Deep Learning with Spectral Contrastive Loss.** Ohad Shamir, Itay Golan, and Eran Malach. Conference on Learning Theory (COLT), 2022.

Thank you for listening.