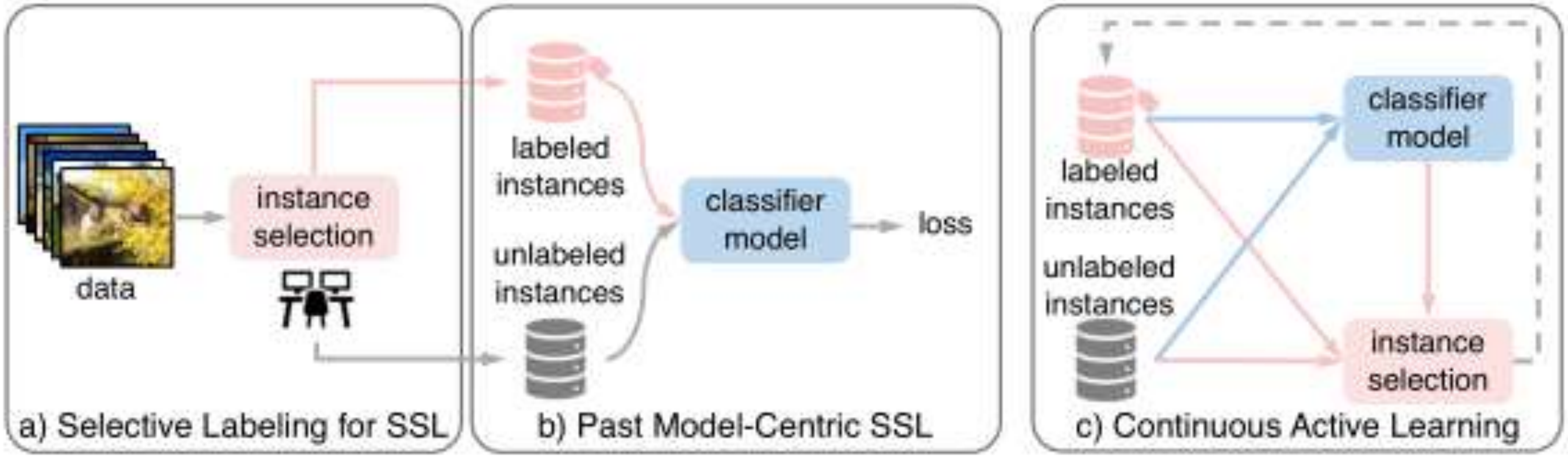
Unsupervised Selective Labeling for More Effective Semi-Supervised Learning

Introduction

- The lower the annotation level, the more important what the labeled instances are to SSL.
- **Random sampling:** Fail to cover all semantic classes
- **Stratified sampling:** Unlabeled instances



- Given only an annotation budget and an unlabeled dataset, select a fixed number of instances for labeling, which way would lead to the best SSL model performance when it is trained on such partially labeled data?

- **Representative**: facilitate label propagation to unlabeled data
- **Diverse**: ensure coverage of the entire dataset
- STEP1: Unsupervised feature learning that maps data into a discriminative feature space.
- STEP2: Select instances for labeling for maximum representativeness and diversity, without or with additional optimization.
- STEP3: Apply SSL to the labeled data and the rest unlabeled data.

Selective Labeling for Semi-supervised Learning

- **Dataset**: unlabeled dataset of n instances
- **Task**: select m ($m \ll n$) instances for labeling, so that a SSL model trained on such a partially labeled dataset produces the best classification performance.

1.Unsupervised Representation Learning

- Obtain lower-dimensional and semantically meaningful features with **unsupervised contrastive learning**
- Map x_i onto a d-dimensional hypersphere with L^2 normalization, denoted as $f(x_i)$

2-1. Unsupervised Selective Labeling (USL)

- We study the relationships between data instances using a **weighted graph**.
- Nodes $\{V_i\}$: instances in the (normalized) feature space $\{f(x_i)\}$
- Edges $\frac{1}{D_{ij}}$: $D_{ij} = \|f(x_i) - f(x_j)\|$

Representativeness: Select Density Peaks

- The K-nearest neighbor density (K-NN) estimation

$$p_{\mathrm{KNN}}(V_i, k) = \frac{k}{n} \frac{1}{A_d \cdot D^d(V_i, V_{k(i)})}$$

- Where $A_d = \pi^{d/2} / \Gamma(\frac{d}{2} + 1)$ is the volume of a unit d-dimensional ball, k(i) instance i's kth nearest neighbor.
- For robustness, we replace it with the average distance

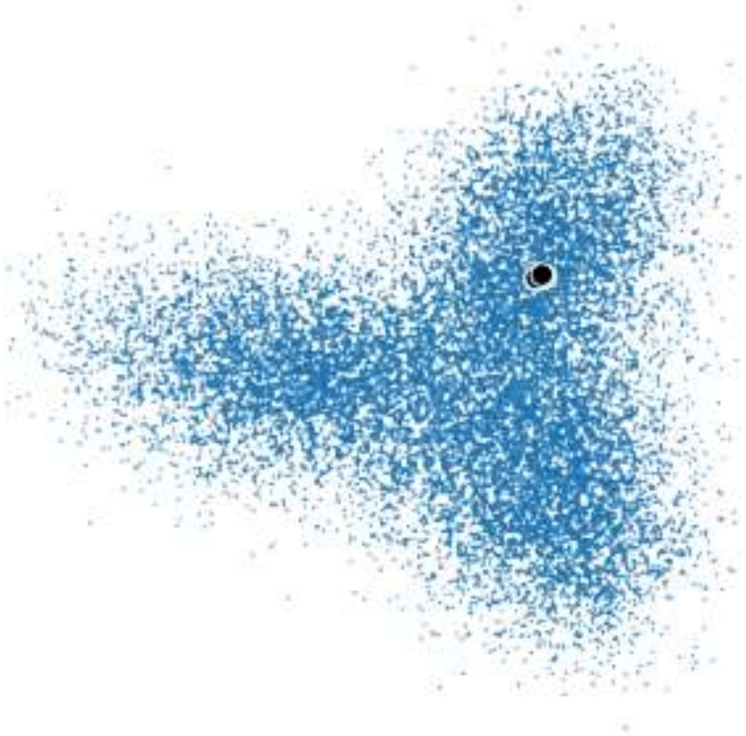
$$\hat{p}_{\mathrm{KNN}}(V_i, k) = \frac{k}{n} \frac{1}{A_d \cdot \bar{D}^d(V_i, k)}, \quad \text{where } \bar{D}(V_i, k) = \frac{1}{k} \sum_{j=1}^{k} D(V_i, V_{j(i)}).$$

Diversity: Pick One in Each Cluster

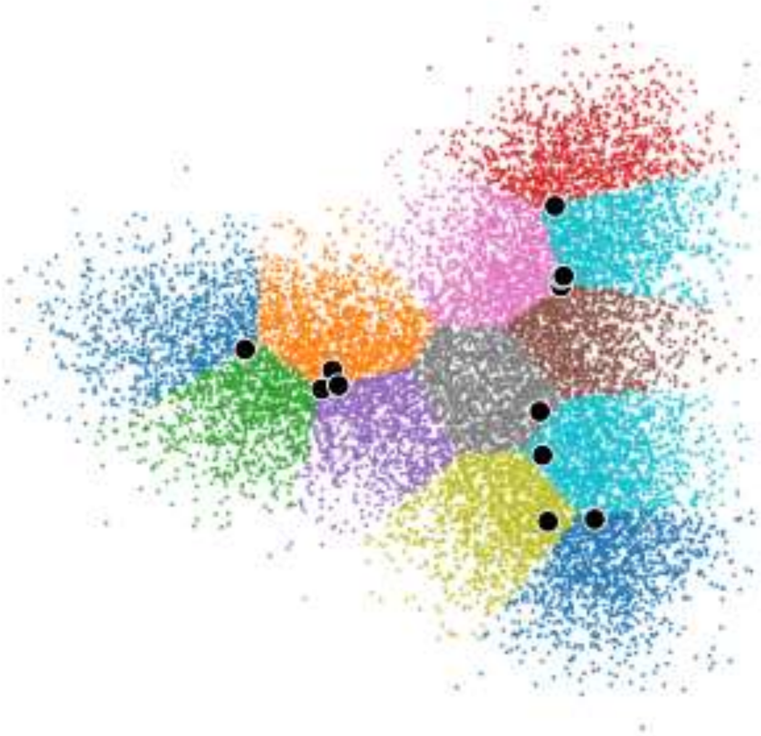
- K-Means clustering that partitions n instances into $m(\leqslant n)$ clusters, with each cluster represented by its centroid c and every instance assigned to the cluster of the nearest centroid.
- we seek m-way node partitioning S = $\{S_1, S_2, \ldots, S_m\}$ that minimizes the within-cluster sum of squares:

$$\min_{\mathbb{S}} \sum_{i=1}^{m} \sum_{V \in S_i} \|V - c_i\|^2 = \min_{\mathbb{S}} \sum_{i=1}^{m} |S_i| \mathrm{Var}(S_i)$$

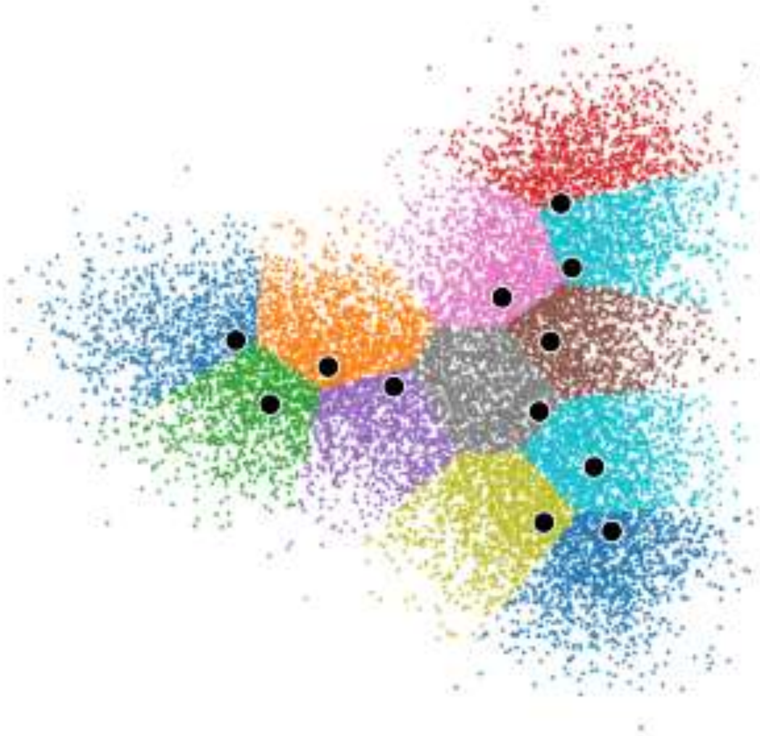
- It is optimized iteratively with EM. We then pick the most representative instance of each cluster.



a) local only



b) local + global



c) local + global + reg.

Regularization: Inter-cluster Information Exchange

- $\hat{\mathbb{V}}^t = \{\hat{V}_1^t, ..., \hat{V}_m^t\}$: the set of *m* instances selected at iteration t.
- For each candidate V_i in cluster S_i, the farther it is away from those in other clusters in $\hat{V}^{t-1}$,the more diversity it creates.
- We thus minimize the total inverse distance to others

$$\mathrm{Reg}(V_i, t) = \sum_{\hat{V}_j^{t-1} \notin S_i} \frac{1}{\|V_i - \hat{V}_j^{t-1}\|^\alpha} \qquad \overline{\mathrm{Reg}}(V_i, t) = m_{\mathrm{reg}} \cdot \overline{\mathrm{Reg}}(V_i, t-1) + (1 - m_{\mathrm{reg}}) \cdot \mathrm{Reg}(V_i, t)$$

- At iteration t, we select instance i of the maximum regularized utility within each cluster

$$U'(V_i, t) = U(V_i) - \lambda \cdot \overline{\mathrm{Reg}}(V_i, t) \qquad U(V_i) = 1/\bar{D}(V_i, k)$$

2-2. Training-Based Unsupervised Selective Labeling (USL-T)

- **Global Constraint via Learnable K-Means Clustering**
- Jointly learn both the cluster assignment and the feature space for unsupervised instance selection
- Suppose that there are C centroids initialized randomly. For instance x with feature f(x), we infer one-hot cluster assignment distribution y(x) by finding the closest learnable centroid c_i, i∈ {1,. . ., C} based on feature similarity s:

$$y_i(x) = \begin{cases} 1, & \text{if } i = \arg\min_{k \in \{1,\ldots,C\}} s(f(x), c_k) \\ 0, & \text{otherwise.} \end{cases}$$

- We predict a soft cluster assignment $\hat{y}(x)$

$$\hat{y}_i(x) = \frac{e^{s(f(x),c_i)}}{\sum_{j=1}^{C} e^{s(f(x),c_j)}}$$

- Minimizing the KL divergence between soft and hard assignments

$$D_{\mathrm{KL}}(y(x) \| \hat{y}(x))$$

- Each instance to become more similar to its centroid (adjust f(x))
- The learnable centroid to become a better representative of instances in the cluster (adjust c)

- Hardening soft assignments has a downside: **Initial mistakes** are hard to correct with later training, degrading performance
- Our solution is to ignore ambiguous instances with maximal softmax scores below threshold τ:

$$L_{\text{global}}(\{x_i\}_{i=1}^{n}) = \frac{1}{n} \sum_{\max(\hat{y}(x_i)) \geq \tau} D_{\text{KL}}(y(x_i) \| \hat{y}(x_i))$$

- As instances are more confidently assigned to a cluster with more training, more instances get involved in shaping both feature f(x) and clusters $\{c_i\}$

- **Our global loss can be readily related to K-Means clustering**
- For τ = 0 and fixed feature f, optimizing L_{global} is equivalent to optimizing K-Means clustering with a regularization term on inter-cluster distances that encourage additional diversity.
- s(.,.) = - L2 distance

$$\{c_i^*\}_{i=1}^{C} = \underset{\{c_i\}_{i=1}^{C}}{\arg\min} \text{ (Main objective + Reg)}$$

where

$$\text{Main objective} = \sum_{x \in \mathcal{X}} ||x - c_{M(x)}||^2$$

$$\text{Reg} = \log \sum_{k=1}^{C} e^{-d(f(x), c_k)} = \log \sum_{k=1}^{C} e^{-||f(x) - c_k||^2}$$

- **Local Constraint with Neighbor Cluster Alignment**
- Soft assignments usually have low confidence scores for most instances at the beginning
- Assigning an instance to the same cluster of its neighbors' in the unsupervisedly learned feature space to prepare confident predictions for the global constraint to take effect

- Two types of collapses:
- (1) Predicting one big cluster for all the instances
- (2) Predicting a soft assignment that is close to a uniform distribution for each instance

- **For one-cluster collapse**
- we adopt a trick for long-tailed recognition and adjust logits to prevent their values from concentrating on one cluster:

$$\hat{P}(z, \bar{z}) = z - \alpha \cdot \log \bar{z}$$

$$\bar{z} = \mu \cdot \sigma(z) + (1-\mu) \cdot \bar{z}$$

- **For even-distribution collapse**
- we use a sharpening function to encourage the cluster assignment to approach a one-hot probability distribution.

$$[P(z, \bar{z}, t)]_i = \frac{\exp(\hat{P}(z_i, \bar{z}_i)/t)}{\sum_j \exp(\hat{P}(z_j, \bar{z}_j/t))}$$

$$L_{\text{local}}(\{x_i\}_{i=1}^{n}) = \frac{1}{n}\sum_{i=1}^{n} D_{\text{KL}}(P(y(x_i'), \bar{y}(x_i'), t)||\hat{y}(x_i)).$$

- We restrict x_i' to x's k nearest neighbors, selected according to the unsupervisedly learned feature prior to training and fixed for simplicity and efficiency.
- Final loss adds up the global and local terms with loss weight λ:

$$L = L_{\text{global}} + \lambda L_{\text{local}}$$

- **Neither one-cluster nor even-distribution collapse is optimal to our local constraint, i.e.,** $P(y(x'), \bar{y}(x'), t) \neq \hat{y}(x)$

$$\hat{P}(z, \bar{z}) = z - \alpha \log \bar{z}$$

$$[P'(\hat{z}, t)]_i = \frac{\exp(\hat{z}_i / t)}{\sum_j \exp(\hat{z}_j / t)}$$

$$P(z, \bar{z}, t) = P'(\hat{P}(z, \bar{z}), t)$$

- **For one-cluster collapse**

$$\begin{aligned} P(z, \bar{z}, t) &= P'(\hat{P}(z, \bar{z}), t) \\ &\approx P'(c\mathbf{1}_d, t) \\ &= \frac{1}{C}\mathbf{1}_d \\ &\neq \hat{y}(x) \end{aligned}$$

For even distribution collapse

$$\begin{aligned} I(z(x'), \bar{z}, t) &= P'(\hat{P}(z(x'), \bar{z}), t) \\ &\approx P'(z(x') - \alpha \log \frac{1}{C}, t) \\ &= P'(z(x'), t) \\ &\neq \hat{y}(x) \end{aligned}$$

- Our USL-T is an **end-to-end unsupervised feature learning** method that directly outputs m clusters for selecting m diverse instances.
- For each cluster, we then select the most **representative** instance, characterized by its highest confidence score $\max \hat{y}(x)$
- Just as USL, USL-T improves model learning efficiency by selecting diverse representative instances for labeling, **without any label supervision**

	MAK	USL-T
Dataset	unlabeled seed training dataset + sampling dataset	unlabeled dataset, without external data
Task	**retrieve an extra set** to enhance self-supervised representation learning	**select partial instances** for labeling, so that a SSL produces the best classification performance
Training Framework	contrastive learning	semi-supervised learning
Principles	**Tailness** $\mathcal{L}^{\mathcal{E}}{}_{\mathrm{CL,i}} = \mathbb{E}_{\theta_{i,1},\theta_{i,2}\sim\Theta}\left(\mathcal{L}_{\mathrm{CL,i}}(\theta_{i,1},\theta_{i,2};\tau,v_i,V^-)\right)$ **Proximity** $D(s^0,s^1) = \frac{1}{\lvert s^1\rvert}\sum_{j\in s^1}\min_{i\in s^0}\Delta(x_i,x_j)$ **Diversity** $H(s^1\cup s^0,S_{all}) = \max_{i\in S_{all}}\min_{j\in s^1\cup s^0}\Delta(x_i,x_j)$	**Representative for each cluster** $\max \hat{y}(x)$ **Diversity** $L_{\mathrm{global}}(\{x_i\}_{i=1}^n) = \frac{1}{n}\sum_{\max(\hat{y}(x_i))\geq\tau} D_{\mathrm{KL}}(y(x_i)\Vert\hat{y}(x_i))$ $L_{\mathrm{local}}(\{x_i\}_{i=1}^n) = \frac{1}{n}\sum_{i=1}^{n} D_{\mathrm{KL}}(P(y(x_i'),\bar{y}(x_i'),t)\Vert\hat{y}(x_i)).$ $L = L_{\mathrm{global}} + \lambda L_{\mathrm{local}}$