Prototypical Contrastive Learning of Unsupervised Representations

Junnan Li, Pan Zhou, Caiming Xiong, Steven C.H. Hoi
Salesforce Research
{junnan.li,pzhou,cxiong,shoi}@salesforce.com

Abstract

This paper presents Prototypical Contrastive Learning (PCL), an unsupervised representation learning method that bridges contrastive learning with clustering. PCL not only learns low-level features for the task of instance discrimination, but more importantly, it implicitly encodes semantic structures of the data into the learned embedding space. Specifically, we introduce prototypes as latent variables to help find the maximum-likelihood estimation of the network parameters in an Expectation-Maximization framework. We propose ProtoNCE loss, a generalized version of the InfoNCE loss for contrastive learning, which encourages representations to be closer to their assigned prototypes. PCL outperforms state-of-the-art instance-wise contrastive learning methods on multiple benchmarks with substantial improvement in low-resource transfer learning. Code and pretrained models are available at: https://github.com/salesforce/PCL.

1 Introduction

Recent advances in unsupervised visual representation learning are largely driven by instance discrimination tasks [1, 2, 3, 4, 5, 6, 7]. Despite their improved performance, instance discrimination methods share a common weakness: the representation is not encouraged to encode the semantic structure of data. This problem arises because instance-wise contrastive learning treats two samples as a negative pair as long as they are from different instances, regardless of their semantic similarity. Thousands of negative samples are generated to form the contrastive loss, leading to many negative pairs that share similar semantics but are undesirably pushed apart in the embedding space.

Clustering methods have also been proposed for deep unsupervised learning [12, 13, 14, 15, 16, 17, 19]. Closest to our work, DeepCluster [18] performs iterative clustering and unsupervised representation learning. It considers cluster assignments as pseudo-labels and optimizes a classification objective, which requires an additional linear classification layer to be frequently re-initialized.

In this paper, we propose *prototypical contrastive learning* (PCL), a new framework for unsupervised representation learning that implicitly encodes the semantic structure of data into the embedding space. A prototype is defined as "a representative embedding for a group of semantically similar instances". We assign several prototypes of different granularity to each instance, and construct a contrastive loss which enforces the embedding of a sample to be more similar to its corresponding prototypes compared to other prototypes. The contributions of this paper can be summarized as:

- We propose prototypical contrastive learning, a unsupervised representation learning method that bridges contrastive learning and clustering. We give a theoretical framework that formulates PCL as an Expectation-Maximization (EM) based algorithm. Specifically, we introduce prototypes as additional latent variables, and estimate their probability in the E-step by performing k-means clustering. In the M-step, we update the network parameters by minimizing our proposed contrastive loss. Under the EM framework, the widely used instance discrimination task can be explained as a special case of prototypical contrastive learning.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

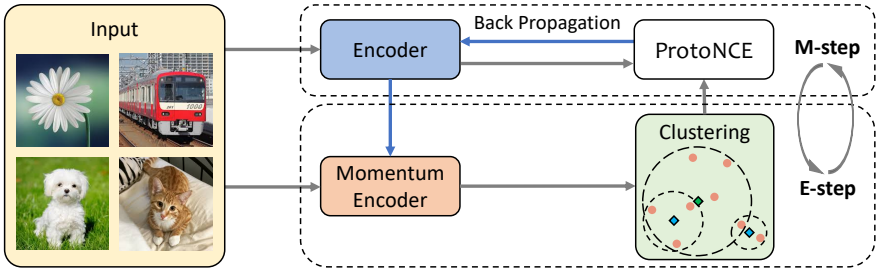

Figure 1: Training framework of Prototypical Contrastive Learning.

- We propose ProtoNCE, a new contrastive loss which improves the widely used InfoNCE by dynamically estimating the concentration for the feature distribution around each prototype.
- PCL outperforms instance-wise contrastive learning on multiple benchmarks with substantial improvements in low-resource transfer learning.

2 Prototypical Contrastive Learning

2.1 Preliminaries

Given a training set $X = \{x_1, x_2, ..., x_n\}$ of n images, unsupervised visual representation learning aims to learn an embedding function f_θ (realized via a DNN) that maps X to $V = \{v_1, v_2, ..., v_n\}$ with $v_i = f_\theta(x_i)$, such that v_i best describes x_i. Instance-wise contrastive learning achieves this objective by optimizing a contrastive loss function, such as InfoNCE [6, 3], defined as:

$$\mathcal{L}_{\text{InfoNCE}} = \sum_{i=1}^{n} -\log \frac{\exp(v_i \cdot v_i'/\tau)}{\sum_{j=0}^{r} \exp(v_i \cdot v_j'/\tau)}, \tag{1}$$

where v_i' is a positive embedding for instance i, and v_j' includes one positive embedding and r negative embeddings for other instances, and τ is a temperature hyper-parameter. In MoCo [3], these embeddings are obtained by feeding x_i to a momentum encoder parametrized by θ', $v_i' = f_{\theta'}(x_i)$, where θ' is a moving average of θ.

2.2 PCL as Expectation-Maximization

Our objective is to find the network parameters θ that maximizes the log-likelihood function of the observed n samples:

$$\theta^* = \arg\max_\theta \sum_{i=1}^{n} \log p(x_i; \theta) \tag{2}$$

We assume that the observed data $\{x_i\}_{i=1}^n$ are related to latent variable $C = \{c_i\}_{i=1}^k$ which denotes the prototypes of the data. In this way, we can re-write the log-likelihood function as:

$$\theta^* = \arg\max_\theta \sum_{i=1}^{n} \log p(x_i; \theta) = \arg\max_\theta \sum_{i=1}^{n} \log \sum_{c_i \in C} p(x_i, c_i; \theta) \tag{3}$$

It is hard to optimize this function directly, so we use a surrogate function to lower-bound it:

$$\sum_{i=1}^{n} \log \sum_{c_i \in C} p(x_i, c_i; \theta) = \sum_{i=1}^{n} \log \sum_{c_i \in C} Q(c_i) \frac{p(x_i, c_i; \theta)}{Q(c_i)} \geq \sum_{i=1}^{n} \sum_{c_i \in C} Q(c_i) \log p(x_i, c_i; \theta), \tag{4}$$

E-step. In this step, we aim to estimate $p(c_i; x_i, \theta)$. To this end, we perform k-means on the features $v_i' = f_{\theta'}(x_i)$ given by the momentum encoder to obtain k clusters. We define prototype c_i as the centroid for the i-th cluster. Then, we compute $p(c_i; x_i, \theta) = \mathbb{1}(x_i \in c_i)$, where $\mathbb{1}(x_i \in c_i) = 1$ if x_i belongs to the cluster represented by c_i; otherwise $\mathbb{1}(x_i \in c_i) = 0$.

M-step. Based on the E-step, we are ready to maximize the lower-bound in eqn.(4).

$$\begin{aligned} \sum_{i=1}^{n} \sum_{c_i \in C} Q(c_i) \log p(x_i, c_i; \theta) &= \sum_{i=1}^{n} \sum_{c_i \in C} p(c_i; x_i, \theta) \log p(x_i, c_i; \theta) \\ &= \sum_{i=1}^{n} \sum_{c_i \in C} \mathbb{1}(x_i \in c_i) \log p(x_i, c_i; \theta) \end{aligned} \tag{5}$$

Under the assumption of a uniform prior over cluster centroids, we have:

$$p(x_i, c_i; \theta) = p(x_i; c_i, \theta)p(c_i; \theta) = \frac{1}{k} \cdot p(x_i; c_i, \theta), \tag{6}$$

We assume that the distribution around each prototype is an isotropic Gaussian, which leads to:

$$p(x_i; c_i, \theta) = \exp\left(\frac{-(v_i - c_s)^2}{2\sigma_s^2}\right) \Big/ \sum_{j=1}^{k} \exp\left(\frac{-(v_i - c_j)^2}{2\sigma_j^2}\right), \tag{7}$$

where $v_i = f_\theta(x_i)$ and $x_i \in c_s$. If we apply ℓ_2-normalization to both v and c, then $(v-c)^2 = 2-2v\cdot c$. Combining this with eqn.(3, 4, 5, 6, 7), we can write maximum log-likelihood estimation as

$$\theta^* = \arg\min_\theta \sum_{i=1}^{n} -\log \frac{\exp(v_i \cdot c_s/\phi_s)}{\sum_{j=1}^{k} \exp(v_i \cdot c_j/\phi_j)}, \tag{8}$$

where $\phi \propto \sigma^2$ denotes the concentration level of the feature distribution around a prototype . Note that eqn.(8) has a similar form as the InfoNCE loss in eqn.(1). Therefore, InfoNCE can be interpreted as a special case of the maximum log-likelihood estimation, where the prototype for a feature v_i is the augmented feature v_i' from the same instance (*i.e.* $c = v'$), and the concentration of the feature distribution around each instance is fixed (*i.e.* $\phi = \tau$).

In practice, we sample r negative prototypes to calculate the normalization term. We also cluster the samples M times with different number of clusters $K = \{k_m\}_{m=1}^{M}$, which enjoys a more robust probability estimation of prototypes. Furthermore, we add the InfoNCE loss to retain the property of local smoothness. Our overall objective, namely **ProtoNCE**, is defined as

$$\mathcal{L}_{\text{ProtoNCE}} = \sum_{i=1}^{n} -\left(\log \frac{\exp(v_i \cdot v_i'/\tau)}{\sum_{j=0}^{r} \exp(v_i \cdot v_j'/\tau)} + \frac{1}{M} \sum_{m=1}^{M} \log \frac{\exp(v_i \cdot c_s^m/\phi_s^m)}{\sum_{j=0}^{r} \exp(v_i \cdot c_j^m/\phi_j^m)} \right). \tag{9}$$

2.3 Concentration estimation

The distribution of embeddings around each prototype has different level of concentration. We use ϕ to denote the concentration estimation, where a smaller ϕ indicates larger concentration. Here we calculate ϕ using the momentum features $\{v_z'\}_{z=1}^{Z}$ that are within the same cluster as a prototype c. The desired ϕ should be small (high concentration) if (1) the average distance between v_z' and c is small, and (2) the cluster contains more feature points (*i.e.* Z is large). Therefore, we define ϕ as:

$$\phi = \frac{\sum_{z=1}^{Z} \|v_z' - c\|_2}{Z \log(Z + \alpha)}, \tag{10}$$

We normalize ϕ for each set of prototypes C^m such that they have a mean of τ.

In eqn.(9), ϕ_s^m acts as a scaling factor on the similarity between an embedding v_i and its prototype c_s^m. With the proposed ϕ, the similarity in a loose cluster are down-scaled, pulling embeddings closer to the prototype. On the contrary, embeddings in a tight cluster have an up-scaled similarity, thus less encouraged to approach the prototype. Therefore, it yields more balanced clusters with similar concentration, which prevents a trivial solution where most embeddings collapse to a single cluster.

3 Experiments

Implementation details. To enable a fair comparison, we follow the same setting as MoCo. We perform training on the ImageNet-1M dataset. A ResNet-50 [31] is adopted as the encoder, whose last fully-connected layer outputs a 128-D and L2-normalized feature. We also experiment with PCL v2 using improvements introduced by [9, 32]. We adopt faiss [33] for efficient k-means clustering.

Low-shot classification. We follow the setup in [34] and train linear SVMs using fixed representations on two datasets: Places205 [35] and PASCAL VOC2007 [36]. We vary the number k of samples per-class and report the average result across 5 independent runs. Table 1 shows the results, in which our method substantially outperforms MoCo.

Method	architecture	VOC07					Places205				
		k=1	k=2	k=4	k=8	k=16	k=1	k=2	k=4	k=8	k=16
Random	ResNet-50	8.0	8.2	8.2	8.2	8.5	0.7	0.7	0.7	0.7	0.7
Supervised	ResNet-50	54.3	67.8	73.9	79.6	82.3	14.9	21.0	26.9	32.1	36.0
Jigsaw	ResNet-50	26.5	31.1	40.0	46.7	51.8	4.6	6.4	9.4	12.9	17.4
MoCo	ResNet-50	31.4	42.0	49.5	60.0	65.9	8.8	13.2	18.2	23.2	28.0
PCL (ours)		**46.9**	**56.4**	**62.8**	**70.2**	**74.3**	**11.3**	**15.7**	**19.5**	**24.1**	**28.4**

Table 1: **Low-shot image classification** with linear SVMs on VOC07 and Places205.

Method	architecture	#pretrain epochs	Top-5 Accuracy 1%	10%
Random [1]	ResNet-50	-	22.0	59.0
Supervised baseline [37]	ResNet-50	-	48.4	80.4
Instance Discrimination [1]	ResNet-50	200	39.2	77.4
Jigsaw [24]	ResNet-50	90	45.3	79.3
SimCLR [9]	ResNet-50-MLP	200	56.5	82.7
MoCo [3]	ResNet-50	200	56.9	83.0
PCL (ours)	ResNet-50	200	**75.3**	**85.6**

Table 2: **Semi-supervised learning** on ImageNet for models finetuned on 1% or 10% of labeled data .

Semi-supervised image classification. Following the setup from [1, 4], we randomly select a subset (1% or 10%) of ImageNet training data (with labels), and fine-tune the self-supervised trained model on these subsets. Table 2 reports the top-5 accuracy on ImageNet validation set. Our method sets a new state-of-the-art under 200 training epochs.

Image classification with linear models. Next, we train linear classifiers on fixed image representations on three datasets: ImageNet, VOC07, and Places205. Table 3 reports the results. PCL outperforms MoCo under direct comparison, which demonstrate the advantage of the proposed prototypical contrastive loss.

Method	architecture (#params)	#pretrain epochs	Dataset ImageNet	VOC07	Places205
DeepCluster [18]	VGG(15M)	100	48.4	71.9	37.9
BigBiGAN [39]	R50 (24M)	–	56.6	–	–
InstDisc [1]	R50 (24M)	200	54.0	–	45.5
MoCo [3]	R50 (24M)	200	60.6	79.2*	48.9*
PCL (ours)	R50 (24M)	200	**61.5**	**82.3**	**49.2**
SimCLR [9]	R50-MLP (28M)	200	61.9	–	–
MoCo v2 [32]	R50-MLP (28M)	200	67.5	84.0*	50.1*
PCL v2 (ours)	R50-MLP (28M)	200	**67.6**	**85.4**	**50.3**

Table 3: **Image classification with linear models.** We report top-1 accuracy. Numbers with * are from released pretrained model; all other numbers are adopted from corresponding papers.

Object detection. Following [34], we train a Faster R-CNN [40] model on VOC07 or VOC07+12, and evaluate on the test set of VOC07. We keep the pretrained backbone frozen to better evaluate the learned representation, and use the same training schedule for both the supervised and self-supervised methods. Table 4 reports the average mAP across three independent runs.

Method	Pretrain Dataset	Architecture	Training data VOC07	VOC07+12
Supervised	ImageNet-1M	Resnet-50-FPN	72.8	79.3
MoCo [3]	ImageNet-1M	Resnet-50-FPN	66.4	73.5
PCL (ours)	ImageNet-1M	Resnet-50-FPN	**71.7**	**78.5**

Table 4: **Object detection** for frozen conv body on VOC using Faster R-CNN.

4 Conclusion

This paper proposes Prototypical Contrastive Learning, a generic unsupervised representation learning framework that finds network parameters to maximize the log-likelihood of the data. Experiments on multiple benchmarks demonstrate the advantage of PCL for unsupervised representation learning.

References

[1] Wu, Z., Y. Xiong, S. X. Yu, et al. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742. 2018.

[2] Ye, M., X. Zhang, P. C. Yuen, et al. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, pages 6210–6219. 2019.

[3] He, K., H. Fan, Y. Wu, et al. Momentum contrast for unsupervised visual representation learning. In *CVPR*. 2020.

[4] Misra, I., L. van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*. 2020.

[5] Hjelm, R. D., A. Fedorov, S. Lavoie-Marchildon, et al. Learning deep representations by mutual information estimation and maximization. In *ICLR*. 2019.

[6] Oord, A. v. d., Y. Li, O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[7] Tian, Y., D. Krishnan, P. Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

[8] Zhuang, C., A. L. Zhai, D. Yamins. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, pages 6002–6012. 2019.

[9] Chen, T., S. Kornblith, M. Norouzi, et al. A simple framework for contrastive learning of visual representations. In *ICML*. 2020.

[10] Tschannen, M., J. Djolonga, P. K. Rubenstein, et al. On mutual information maximization for representation learning. In *ICLR*. 2020.

[11] Saunshi, N., O. Plevrakis, S. Arora, et al. A theoretical analysis of contrastive unsupervised representation learning. In *ICML*, pages 5628–5637. 2019.

[12] Xie, J., R. B. Girshick, A. Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487. 2016.

[13] Yang, J., D. Parikh, D. Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, pages 5147–5156. 2016.

[14] Liao, R., A. G. Schwing, R. S. Zemel, et al. Learning deep parsimonious representations. In *NIPS*, pages 5076–5084. 2016.

[15] Yang, B., X. Fu, N. D. Sidiropoulos, et al. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *ICML*, pages 3861–3870. 2017.

[16] Chang, J., L. Wang, G. Meng, et al. Deep adaptive image clustering. In *ICCV*, pages 5880–5888. 2017.

[17] Ji, X., J. F. Henriques, A. Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, pages 9865–9874. 2019.

[18] Caron, M., P. Bojanowski, A. Joulin, et al. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 139–156. 2018.

[19] Caron, M., I. Misra, J. Mairal, et al. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*. 2020.

[20] Pathak, D., P. Krähenbühl, J. Donahue, et al. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544. 2016.

[21] Zhang, R., P. Isola, A. A. Efros. Colorful image colorization. In *ECCV*, pages 649–666. 2016.

[22] Zhang, R., P. Isola, A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, pages 1058–1067. 2017.

[23] Doersch, C., A. Gupta, A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430. 2015.

[24] Noroozi, M., P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84. 2016.

[25] Dosovitskiy, A., J. T. Springenberg, M. A. Riedmiller, et al. Discriminative unsupervised feature learning with convolutional neural networks. In *NIPS*, pages 766–774. 2014.

[26] Gidaris, S., P. Singh, N. Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*. 2018.

[27] Caron, M., P. Bojanowski, J. Mairal, et al. Unsupervised pre-training of image features on non-curated data. In *ICCV*, pages 2959–2968. 2019.

[28] Zhang, L., G. Qi, L. Wang, et al. AET vs. AED: unsupervised representation learning by auto-encoding transformations rather than data. In *CVPR*. 2019.

[29] Deng, J., W. Dong, R. Socher, et al. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. 2009.

[30] Ross, B. C. Mutual information between discrete and continuous data sets. *PloS one*, 9(2), 2014.

[31] He, K., X. Zhang, S. Ren, et al. Deep residual learning for image recognition. In *CVPR*, pages 770–778. 2016.

[32] Chen, X., H. Fan, R. Girshick, et al. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[33] Johnson, J., M. Douze, H. Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

[34] Goyal, P., D. Mahajan, A. Gupta, et al. Scaling and benchmarking self-supervised visual representation learning. In *ICCV*, pages 6391–6400. 2019.

[35] Zhou, B., À. Lapedriza, J. Xiao, et al. Learning deep features for scene recognition using places database. In *NIPS*, pages 487–495. 2014.

[36] Everingham, M., L. V. Gool, C. K. I. Williams, et al. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[37] Zhai, X., A. Oliver, A. Kolesnikov, et al. S4l: Self-supervised semi-supervised learning. In *ICCV*, pages 1476–1485. 2019.

[38] Miyato, T., S. Maeda, M. Koyama, et al. Virtual adversarial training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(8):1979–1993, 2019.

[39] Donahue, J., K. Simonyan. Large scale adversarial representation learning. In *NeurIPS*, pages 10541–10551. 2019.

[40] Ren, S., K. He, R. B. Girshick, et al. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99. 2015.