Improving ELSA and adding a theoretical framework

A Rameswar Patro & Aaditya Vicram Saraf

School of Mathematical Sciences

National Institute of Science Education and Research Bhubaneswar

P.O. Jatni, Khurda 752050, Odisha, India

arameswar.patro@niser.ac.in & aadityavicram.saraf@niser.ac.in

April 20, 2024

Abstract

In the context of an unlabelled image dataset and a reference image, the objective is to identify images similar to the reference. While a common approach involves labeling images as positive (similar) or negative (dissimilar), this method becomes inefficient when dealing with imbalanced datasets where negative samples vastly outnumber positive ones. To address this challenge, we propose leveraging ELSA (Explorative Latent Self-supervised Active Search Algorithm). ELSA comprises three key elements: a sophisticated Random Sampler, an Iterated Nearest Neighbor Search, and a linear head enhanced by active learning. Our goal is to enhance ELSA by refining the sampler's structure and providing theoretical insights into its functionality.

1 Introduction

Given a set of unlabelled images $\mathbb{D}$ and a reference image, we want to find all the images similar to the reference image. Let's call the reference image to be seed. An image in $\mathbb{D}$ is called positive (or negative) if it is similar (or dissimilar) to the seed. One way to find all the positive samples is by labelling everything in $\mathbb{D}$. But the ratio of positive samples to the negative samples can be as bad as 1:1000. In those cases the above method will be inefficient.

The algorithm ELSA uses Self-Supervised learning to transform the dataset to a specific space (latent space) where images with similar characteristics are clustered together. Then it uses Random Sampler, and Nearest Neighbour search in the latent space to find out points similar to the seed. The Random

Sampler (RandS) selects points from random clusters using a single layer of neural network (which is trained on the data from the previous iterations) , while Nearest Neighbour Search (NN) explores nearby points to uncover entire clusters.

During the early stages, RandS and NN operate independently. However, with increasing iterations, RandS improves its predictions, potentially labeling most samples as positive, leading to clustering issues where all points may belong to one or two clusters. Although NN can discover other points within this cluster, it is essential for points to be from distinct clusters. To address this, we propose modifying the random sampler to predict the most probable positive points.

2 Related works

Self Supervised Learning: Self-supervised learning is a form of unsupervised learning that plays a crucial role in advancing image processing capabilities. By leveraging unstructured and unlabeled data, self-supervised learning facilitates the development of generic artificial intelligence systems. This approach allows images to learn from their visual features autonomously, without the need for human-generated labels. Self-supervised learning encompasses various techniques such as contrastive learning and transfer learning, enabling the extraction of meaningful representations from data without explicit supervision. Through self-supervised learning, models can be trained to predict hidden parts of input data based on visible information, fostering the acquisition of robust and informative features for downstream tasks in computer vision applications. In our context, self-supervised learning serves as a transformative tool to map the image set into a latent space. Noteworthy self-supervised algorithms that could be considered include VICReg (Bardes et al., 2022), Barlow Twins (Zbontar et al., 2021), MoCo (He et al., 2020), MoCo V2 (Chen et al., 2020b), ObOW (Gidaris et al., 2021), SimSiam (Chen & He, 2020), SwAV (Caron et al., 2019), and SwAV-multi-crop (Caron et al., 2019).

Farthest Point Sampling: Farthest point sampling is a technique used to select a subset of points that are maximally spread out from each other. This method aims to ensure diversity and coverage within the selected points by iteratively choosing the point that is farthest from the existing set. By prioritizing points that are distant from one another, farthest point sampling helps in creating representative subsets that capture the overall distribution or characteristics of the data without redundancy. We aim to employ it in our sampler so that the random sampler would choose points far from each other avoiding taking points from same cluster.

3 Some attempts to change the Random Sampler

The Random Sampler of ELSA is not very effective in choosing points that are far from each other. This may affect the performance in the later stages of the algorithm as the linear head becomes better and better with each iteration.

Greedy Sampler(GreedyS): The proposed modification to the Random Sampler of ELSA aims to enhance point selection efficiency by introducing the Greedy Sampler (GreedyS) approach. Instead of selecting points collectively, GreedyS suggests choosing points individually based on confidence levels determined by the linear head. Subsequently, the selected point is labeled by an Oracle, followed by an exhaustive Nearest Neighbor search to identify all points within its cluster. This method prioritizes confident point selection to prevent distant point selections, thereby optimizing sampler performance as the algorithm progresses. The concept of Farthest Point Sampling (FPS) aligns with this approach, as it involves iteratively selecting a point followed by Nearest Neighbor Search. So the new point on the next iteration would not belong to any cluster of previously visited points. By leveraging FPS principles, it is possible to enhance point selection strategies within ELSA while minimizing computational complexity and improving sampling performance.

Farthest point Sampler(FarS): The Farthest Point Sampler (FarS) method enhances point sampling by incorporating farthest point sampling and the linear head. It aims to maximize a function $g : \mathbb{D}^n \to \mathbb{R}$ defined as the sum of pairwise distances between points and the confidence values output by the linear head.

$$g(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} ||x_i - x_j|| + \sum_{i=1}^{n} f_c(x_i)$$

where f_c is the output of the linear head. By maximizing both confidence values and inter-point distances, FarS identifies points of interest that are crucial for the algorithm. These selected points are then labeled by an Oracle and subjected to a nearest neighbor search if identified as positive. FarS leverages Farthest Point Sampling principles to streamline the process of finding positive samples efficiently, reducing the number of iterations required. This approach optimizes sampling by emphasizing both point confidence and spatial distribution, enhancing the effectiveness of point selection within ELSA.

Grid Sampler(GridS): The Grid Sampler (GridS) method enhances point sampling by transforming the latent space into a lower-dimensional space using Principal Component Analysis (PCA). This transformation facilitates the division of the space into rectangular grids using hyperplanes. Subsequently, GridS identifies the most confident point within each grid and selects the maximum confident points among them. By employing this approach, GridS ensures that all samples are positioned at a distance from each other, promoting spatial diversity. Moreover, by reducing the dimensionality through grid-based search,

GridS effectively manages computational complexity as the number of search grids grows exponentially with increasing dimensions. This method optimizes point selection within ELSA by leveraging PCA for dimension reduction and grid-based sampling to enhance spatial distribution and sampling efficiency.

4 Evidence behind ELSA

In this section we will just define some terms that will be helpful for providing evidence behind working of ELSA. First we will measure the effectiveness of the algorithm by Labelling efficiency. The Labelling Efficiency (L_e) is defined as the ratio of the labelled positive points to the total number of labelled points.

Next we define what a cluster is. A set S of points is called a cluster if it is same as the output of iterated Nearest Neighbour Search algorithm done on any single point in S. Further we analyse what issues can cause an error while running ELSA. Based on our understanding, majorly 3 factors contribute to this error. They are as follows :

1. **Latent Space Error (χ):** This is the VICReg space error for our model, and it basically tells how accurate the latent space representation is. In a perfect scenario, we only get one compact cluster. We will assume χ to be a fixed value depending on the dataset which is to be calculated empirically, and we failed to predict much about this error. Hence we may assume this to be a fixed value for a dataset. For further understanding we may refer to [2].

2. **Oracle Error :** This accounts for errors made by the oracle in labelling. This is comprises of 2 errors $\varepsilon_{(+)}$ and $\varepsilon_{(-)}$ which are the positive and negative errors respectively. Positive error is an error made by the oracle where he labels a positive point as negative and negative error is labelling of a negative point as positive. Since we have a huge number of positive samples, we assume that $\varepsilon_{(-)} << \varepsilon_{(+)}$ to ensure working of the algorithm.

3. **MLP Error :** This depends on the performance of MLP in each iteration. Since this may vary hugely depending on the dataset, to simplify our analysis, we make an assumption that MLP is perfect in higher confidence ranges and gets higher accuracy in lower confidence ranges as we increase the number of iterations.

5 Linearity of Oracle

In this section, we put minimal assumptions on MLP error and latent space error. We observe that the oracle's error turns out to be linear, and hence after this section we consider Oracle to be ideal.

Define the following :

- $L^+_{NN}(n)$ = Number of positive samples selected by an NN iteration in epoch n. We use a kNN model, hence

$$L^+_{NN}(1) = k(1-\chi).$$

- $L^-_{NN}(n)$ = Number of negative samples selected by an NN iteration in epoch n.

$$L^-_{NN}(1) = k\chi.$$

- $E_R(n)$ = The fraction of negative points picked up by the RandS sampler from R points in the nth iteration.

- $L^+_R(n)$ = Number of positive samples selected by the random sampler in epoch n. We are taking R random samples in the nth epoch, hence

$$L^+_R(1) = R(1 - E_R(1)).$$

- $L^-_R(n)$ = Number of negative samples selected by the random sampler in epoch n. We are taking R random samples in the nth epoch, hence

$$L^+_R(1) = RE_R(1).$$

With these notations we get the following table for any iteration of nearest neighbour component.

	Labelled Positive	Labelled Negative
True Positive	$k(1-\chi)(1-\varepsilon_{(+)})$	$k(1-\chi)\varepsilon_{(+)}$
True Negative	$k\chi\varepsilon_{(-)}$	$k\chi(1-\varepsilon_{(-)})$

Hence the total number of Labelled positives is $k(1-\chi)(1-\varepsilon_{(+)}) + k\chi\varepsilon_{(-)}$. By our assumption on oracle error, we get

$$k\chi\varepsilon_{(-)} << k(1-\chi)(1-\varepsilon_{(+)}).$$

So the total number of labelled positives can be approximated to

$$L^+_{NN}(n) = k(1-\chi)(1-\varepsilon_{(+)}).$$

Similarly we get the following table for nth iteration of Random Sampler component.

	Labelled Positive	Labelled Negative
True Positive	$R(1-E_R(n))(1-\varepsilon_{(+)})$	$R(1-E_R(n))\varepsilon_{(+)}$
True Negative	$RE_R(n)\varepsilon_{(-)}$	$RE_R(n)(1-\varepsilon_{(-)})$

Hence the total number of Labelled positives is

$$R(1 - E_R(n))(1 - \varepsilon_{(+)}) + RE_R(n)\varepsilon_{(-)}.$$

By our assumption on oracle error, we get

$$RE_R(n)\varepsilon_{(-)} << R(1 - E_R(n))(1 - \varepsilon_{(+)}).$$

So the total number of labelled positives can be approximated to

$$L_R^+(n) = R(1 - E_R(n))(1 - \varepsilon_{(+)}).$$

After N steps of nearest neighbours and Random Sampler, the total number of labelled positives is equal to

$$\begin{aligned}\sum_{n=1}^{N}(L_R^+(n) + L_{NN}^+(n)) &= \sum_{n=1}^{N}(k(1-\chi)(1-\varepsilon_{(+)}) + R(1 - E_R(n))(1 - \varepsilon_{(+)})) \\ &= (kN(1-\chi) + R(N - \sum_{n=1}^{N} E_R(n)))(1 - \varepsilon_{(+)})\end{aligned}$$

Total labelled points is equal to $(R + k)n$. Hence labelling efficiency with oracle error is equal to

$$L'_e = \frac{\left(kN(1-\chi) + R(N - \sum\limits_{n=1}^{N} E_R(n))\right)(1 - \varepsilon_{(+)})}{(R+k)N} = L_e(1 - \varepsilon_{(+)}).$$

This shows that the oracle error affects labelling efficiency on a linear scale.

Note : Even if χ is assumed to be varying with each iteration, which might be a more likely case, the Oracle's linear dependence is unaffected. In this situation, we will get our expression to be

$$L'_e = \frac{\left(k(N - \sum\limits_{n=1}^{N} \chi(n)) + R(N - \sum\limits_{n=1}^{N} E_R(n))\right)(1 - \varepsilon_{(+)})}{(R+k)N} = L_e(1 - \varepsilon_{(+)}).$$

6 Error Analysis

Assumptions :

- We have assumed that the confidence assigned by the MLP has error of a linear factor. It means that if we select points from a confidence range of (a,b), we get at least $\alpha\frac{a+b}{2}$ positive points.
- VICReg error (χ) is a constant throughout the performance.

In the above setting, the value of E_R will be

$$E_R = 1 - \alpha \frac{a+b}{2}.$$

This gives us that labelling efficiency (L_e) to be

$$\begin{aligned} L_e &= \frac{\left(kN(1-\chi) + R(N - \sum_{n=1}^{N} E_R\right)}{(R+k)n} \\ &= \frac{\left(kN(1-\chi) + R(N - \sum_{n=1}^{N} (1 - \alpha \frac{a+b}{2})\right)}{(R+k)n} \\ &= \frac{\left(kN(1-\chi) + RN\alpha \frac{a+b}{2}\right)}{(R+k)N}. \end{aligned}$$

7 Further plans

Our previous theoritical aspirations (quoted in the below box) have been more or less achieved.

> In our pursuit to enhance ELSA's performance, we have explored alternative sampling methods to replace RandS and are currently implementing them on our datasets to evaluate potential performance improvements. Concurrently, we are actively engaged in developing the theoretical framework of ELSA to deepen our understanding of its underlying principles and mechanisms. By incorporating diverse sampling strategies like GreedySampler, Farthest Point Sampling (FPS), and grid-based sampling, inspired by the mentioned resources, we aim to optimize point selection within ELSA. This iterative process of experimentation and theoretical refinement underscores our commitment to advancing ELSA's efficacy through innovative sampling techniques and a robust theoretical foundation.

Furthermore, currently we are rectifying the pseudocodes in the original ELSA paper and we have simplified the notations.

References

1. *Bardes, A., Ponce, J., & LeCun, Y. (2021). VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning (Version 3). arXiv.*
2. *Shwartz-Ziv, R., Balestriero, R., Kawaguchi, K., Rudner, T. G. J., & LeCun, Y. (2023). An Information-Theoretic Perspective on Variance-Invariance-Covariance Regularization (Version 2). arXiv.*
3. *Li, J., Zhou, J., Xiong, Y., Chen, X., & Chakrabarti, C. (2022). An Adjustable Farthest Point Sampling Method for Approximately-sorted Point Cloud Data (Version 1). arXiv.*