Robust Contrastive Learning

Rana Muhammad Shahroz Khan [1] **Zhanqi Zhu** [1] **Du Duong** [1] **Aryan Katneni** [1]

Abstract

Clustering is recognized as a task of grouping similar data points without having any of their label information. In this paper we propose an online, robust Deep Clustering method that utilizes Contrastive Learning coupled with representation learning on Unit Hyper spheres. To be specific, the architecture uses a cluster-level projection head and a instance-level project head to cluster the different unlabelled images by generating pairwise pseudo labels from the different augmentations. We employ these contrastive losses to learn better aligned representations of our data. To better utilize the full unit hyper sphere we also make use of a uniformity loss which allows us to use the complete hyper sphere to learn linearly separable representations. The test results on MNIST dataset look promising where our linear classifier could achieve an accuracy of 87% on the learned representation from our method, compared to 78% from a model only utilizing the contrastive loss. We go on to empirically verify how our method yields robust linearly separable representations compared to the vanilla contrastive loss.

1. Introduction

Clustering is a fundamental task in unsupervised learning where we try to group similar data points together without the need for their corresponding labels. Clustering has had huge success and found its need in different applications like anomaly detection (Ahmed et al., 2016), data mining tasks (Berkhin, 2006), image segmentation tasks (Coleman & Andrews, 1979; Chuang et al., 2006) and even medical analysis (Bruse et al., 2017) and so has been active area of research in Machine Learning for quite some time. Historically, clustering algorithms in the space of traditional machine learning like k-means (Coates & Ng, 2012) and Spectral Clustering (Ng et al., 2001) have yielded great results on lower dimensional inputs, however, they have failed wonderfully when it comes to high-dimensional input like images.

Following the recent success of deep learning with high dimensional inputs the current research in clustering has been directed by such deep neural networks. With the advent of better representation learning modes like Auto-encoders (Kingma & Welling, 2013; Bengio et al., 2006), the steering wheel for most modern clustering algorithms has been turned towards deep clustering (Caron et al., 2018) based approaches. These approaches involve grouping features using k-means networks iteratively, however, this iterative nature of the method gives rise to accumulation of error which damages the performance on the validation set considerably. Moreover, all such methods operate in an offline manner i.e, they require all data that needs to be clustered at train time and hence are not very suitable for deployment in large-scale systems where the data comes and goes while in operation (online learning).

To overcome this problem a Contrastive Clustering (Caron et al., 2018) algorithm was proposed where in the model was trained using a batch of data points/images instead of all the data. This made the architecture suitable for online clustering and deployment in large scale systems like recommender systems, anomaly detection as well as video monitoring systems that require image segmentation tasks to be performed in an online manner. The paper builds upon the SimCLR (Chen et al., 2020) framework with 2 contrastive heads instead of 1 based on the observations of ”representation as label”. With the help of a pairwise pair construction backbone, an image is augmented to generate 2 images using the same augmentation family, passed on to a deep convolution network and spits out 2 representations (one for each augmented image). Utilizing the row space for instance level contrastive head, and column space for the cluster level contrastive head, 2 MLPs are utilized to achieve the task of clustering.

However, problems arose when we studied this method in detail and visualized the unit hyper-sphere representation learned by this contrastive clustering method. The unit hyper-sphere was not being utilized completely, leaving behind a lot of empty space. Hence we would like to learn

[*] Equal contribution [1] Department of Computer Science, Vanderbilt University, TN, US. Correspondence to: Rana Muhammad Shahroz Khan <rana.muhammad.shahroz.khan@vanderbilt.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

a representation that was more aligned with the data and utilizes the unit hyper-sphere in greater efficiency.

Wang and Isola (Wang & Isola, 2020) empirically verify that there is a need for uniformity alongside alignment for better representation learning in the unit hyper-sphere. The need for uniformity is hypothesized to make the clusters in the representations more linearly separable allowing for robust learning. They also show that the contrastive InfoNCE (Oord et al., 2018) loss can also be reduced to the sum of uniformity and alignment losses adding more theoretical background to their empirically verifiable results.

In this paper we introduce a novel loss for robust contrastive clustering for highly non-linear, high dimensional data like images, that can be deployed on large scale monitoring and recommender systems as it is fully capable of online learning and is robust to biases in data. Our contributions can be summarized as below :

- A new contrastive clustering loss that utilizes an instance level contrastive loss, cluster level contrastive loss and a newly added uniformity loss, which can allow us to align our cluster representations as well as is capable of utilizing the vast space of the unit hyper-sphere.
- We show that this new loss can give us better linearly separable representations.
- We also show empirically that the our method is robust to biases and errors found in data like mislabeling of classes and incorrect number of classes in data. Our method even when trained on incorrect number of classes, can still recover a representation that clearly follows the ground truth.

The results for all the experiments are shown below alongside their testing method in Section 5.

Some limitations that can apply to our method are the lack of widespread testing on other data sets other than MNIST as well as sub-optimal hyper parameter calibration due to the interest in time and lack of enough compute resources.

2. Related Work

2.1. Contrastive Learning

Contrastive learning is an unsupervised representation learning method (Chen et al., 2020). A data sample undergoes two augmentations that randomly transform the data. The resulting views are known as a positive pair. Then, a neural network encoder $f(\cdot)$ extracts the important features from the augmented samples. Next, a neural network projection head $g(\cdot)$ transforms the encoded representation to a space where the contrastive loss can be applied. The goal of the contrastive loss is to minimize loss for the positive pairs and maximize loss everywhere else. Our method utilizes contrastive learning to find good representations for images.

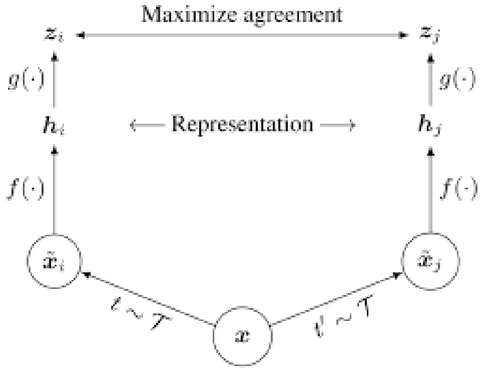

Figure 1. Simple framework for contrastive learning

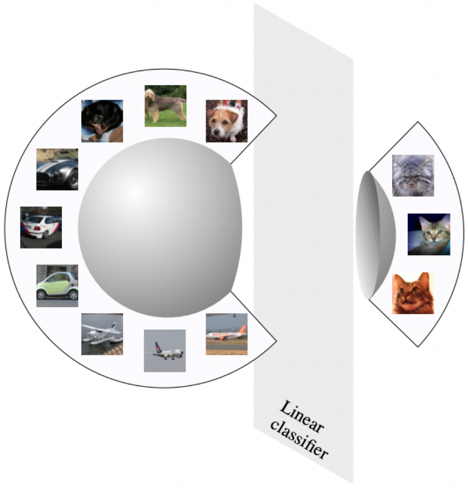

Figure 2. Well-clustered classes on a hypersphere are linearly separable

2.2. Deep Clustering

When trained with supervision, a convnet can map images to good general-purpose features (Caron et al., 2018). The convnet requires labels in order to learn the parameters of the neural network. Deep clustering utilizes the discriminative power of convnet combined with clustering to learn features without supervision. As shown in Figure 6, clustering is applied to the output of the convnet. These clusters are then used as pseudo-labels for classification, which then are used to minimize loss.

2.3. Unit Hyper sphere Representation Learning

Wang and Isola introduce *alignment* and *uniformity* as desirable properties for the distribution of features on the unit

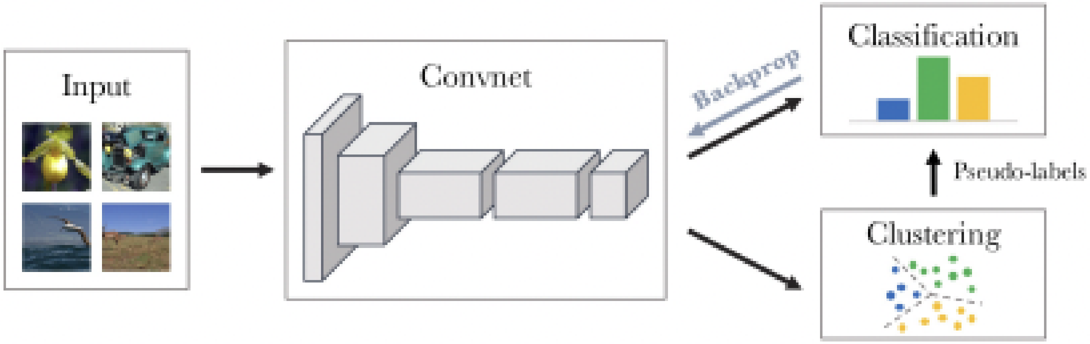

Figure 3. Illustration of deep clustering method

hypersphere. When these properties are maximized, then downstream task performance is greatly increased. Alignment implies that positive pairs have features that are close. Uniformity simply says that the distribution of features on the unit hypersphere should be uniform throughout, preserving as much information as possible. Maximizing these properties allows classes to be linearly separable, as shown in Figure 5. By adding uniformity and alignment losses in addition to the contrastive loss, the performance of the traditional contrastive learning method can be improved.

3. Contrastive Clustering and Uniformity

3.1. Contrastive Clustering

Contrastive Clustering (CC) (Li et al., 2021) is a one-stage online deep clustering method which performs instance- and cluster-level contrastive learning independently via rows (instance) and columns (cluster) of feature matrices.

The model of CC is constructed with three jointly learned components: a pair construction backbone (PCB), an instance-level contrastive head (ICH), and a cluster-level contrastive head (CCH). By optimizing the instance- and cluster- level contrastive loss, the model learned representations and and clusters simultaneously.

- **Pair Construction Backbone** : This component uses data augmentation to construct data pairs, and pass them through a deep neural network to generate their features. Given a input x_i, two stochastic data augmentation samples T^a, T^b, sampled from the same family of augmentations T are applied to x_i, resulting in two correlated samples $x_i^a = T^a(x_i)$ and $x_i^b = T^b(x_i)$. Then, a deep neural network f is used to extract augmentation samples x_i^a and x_i^b, into two corresponding features $h_i^a = T^a(x_i)$ and $h_i^b = T^b(x_i)$.

- **Instance-Level Contrastive Head**: Contrastive learning aims to maximize the similarity between positive pair and minimizing the similarity between negative pairs. Since no prior label is available for this clustering task, positive pair are defined as the samples augmented from the same instance, while negative pairs are define as different samples augmented from the same or different instance. During training, CC is given a mini-batch input of size *N*. It generates 2*N* data samples $\{x^a, ..., x_N^a, x_1^b, ..., x_N^b\}$ through two data augmentations. Given a sample x_i, there are 1 positive pair $\{x_i^a, x_i^b\}$, and 2*N*-2 negative pairs.
CC does not directly conduct contrastive learning on the feature matrices. Instead, a two-layer MLP g_I is used to map the two features into $z_i^a = g_I(h_i^a)$, and $z_i^b = g_I(h_i^b)$. The instance-level contrastive loss is applied here via pair-wise similarity measured by cosine distance:

$$s(z_i^{k_1}, z_j^{k_2}) = \frac{(z_i^{k_1})(z_j^{k_2})^T}{\|z_i^{k_1}\|\|z_j^{k_2}\|}$$

where $k_1, k_2 \in \{a, b\}$ and $i, j \in [1, N]$. The cross entropy loss of a given sample x_i^a to optimize pair-wise similarities is in the form of

$$l_i^a = -\log \frac{exp(\frac{S(z_i^a, z_i^b)}{\tau})}{\sum_j^N exp(\frac{S(z_i^a, z_j^a)}{\tau_I}) + exp(\frac{S(z_i^a, z_j^b)}{\tau_I})}$$

where τ_I is the instance-level temperature parameter. The instance-level contrastive loss L_{ins} is computed over every augmented samples since the aim is to maximize the similarity between all possible pairs.

$$L_{ins} = \frac{1}{2N}\sum_{i=1}^{N}(l_i^a + l_i^b)$$

- **Cluster-Level Contrastive Head**: CCH aims to maximize the similarity between each columns of feature matrix while minimizing the entropy of cluster assignment probability. Through a MLP g_C independent to

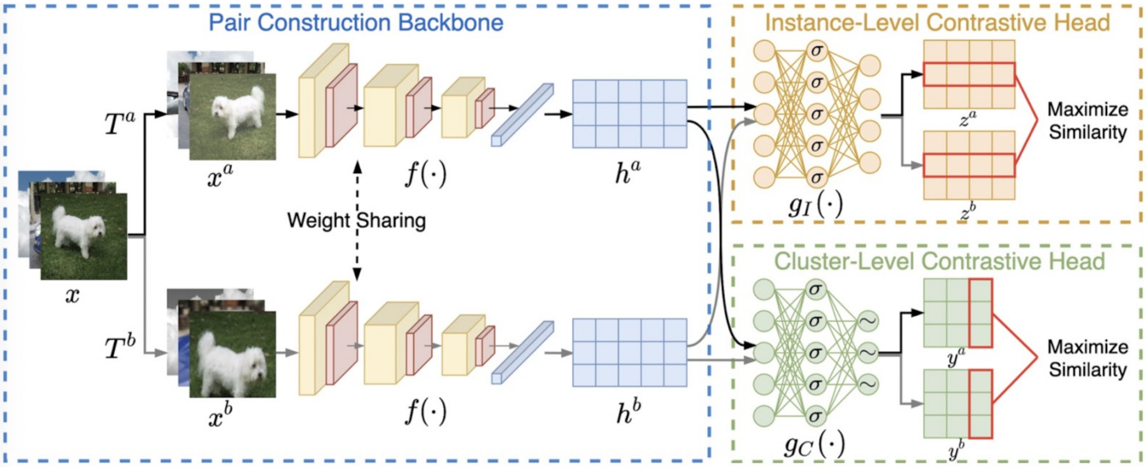

Figure 4. Architecture of Contrastive Clustering. Taken from (Li et al., 2021)

g_I in ICH, feature h_i^a is projected into M-dimensional space via $\hat{y}_i^a = g_C(h_i^a)$, where M is the number of clusters. Let $Y^a \in R^{N \times M}$ be the output of CCH for a mini-batch of data augmentation inputs, $\hat{y}_i^a$ is the soft label of sample $\hat{x}_i^a$ i-th row of Y^a.

Let y_i^a be the ith column of Y^a, which is the representation of cluster *i* under the first augmentation *a*. The positive pair is defined as $\{y_i^a, y_i^b\}$, while the rest 2*M*-2 pairs are negative.

CCH measures the similarity between cluster pairs via cosine similarity as ICH. The loss that distinguish y_i^a with all other clusters except y_i^b is

$$\hat{l}_i^a = -\log \frac{exp(\frac{S(y_i^a, y_i^b)}{\tau_C})}{\sum_j^N exp(\frac{S(y_i^a, y_j^a)}{\tau_C}) + exp(\frac{S(y_i^a, y_j^b)}{\tau_C})}$$

where τ_C is the cluster-level temperature paramter. Aside from maximizing the similarity between clusters, CCH also need to reduce entropy of cluster assignment probabilities so that the assignment is more ”certain”.

$$H(Y) = \sum_{k=1}^{M} [P(y_i^a) log P(y_i^a) + P(y_i^b) log P(y_i^b)]$$

Finally, the cluster-level contrastive loss is

$$L_{clu} = \frac{1}{2M} \sum_{i=1}^{M} (\hat{l}_i^a + \hat{l}_i^b) - H(Y)$$

- **Overall Objective Function**: During training of CC, two heads ICH and CCH are simultaneously optimized. The overall contrastive loss function is a sum of both instance- and cluster- level contrastive loss

$$L_{cont} = L_{ins} + L_{clu}$$

while a hyper-parameter could be used to balance the weight of two losses.

3.2. Uniformity in Unit Hyper sphere

Uniformity is an essential key property related to contrastive loss. It prefers a feature distribution that preserves maximal information, therefore allowing the model to learn a better representation utilized hypersphere by adding a uniformity loss.

The uniformity loss is defined as the logarithm of the average pairwise Gaussian potential.

$$L_{uniformity}(f; t) = log E_{x,y \ p_{data}}[G_t(u, v)]$$

where $G_t(u, v)$ is the Gaussian potnetial kernel (also known as the Radial Basis Function (RBF) kernel)

$$G_t(u, v) = e^{-t\|u-v\|_2^2}, t > 0$$

4. Robust Constrastive Clustering

After trying out contrastive clustering and uniformity losses separately we wanted ability to combine both of these ideas into one idea. Contrastive clustering, although an online clustering method that works great on most data sets, fails to exploit the hyper-sphere representation fully. Most of the clusters lied upon each other that allowed for much empty and not utilized space in the hyper-sphere.

On the other side the uniformity and the alignment loss do not alone give good comparable performance when it comes to clustering as these two loss terms were derived clearly from the InfoNCE contrastive loss which has nothing to do with the 2 instance and cluster level projection contrastive losses.

Some problems we could think of in regards to the vanilla CC method were that it was very much dependent on the correct number of classes,K, to generate good somewhat linearly separable cluster representations. However, with a little induced twist in K the representations ended up being not very linear. Hence the need for correct K for CC method, and a slight variation in it could damage our clustering accuracy. Outside the academic experiments, in most unlabelled, raw data, we do not have access to the correct number of classes in the data. However, there are some methods like k-means coupled with silhoutte method or elbow method to infer the number of classes, k-means alone does not work very well in an online setting and with complex high dimensional data sets. Other than just the robustness to K, we also noticed that the CC does not fully exploit the hyper-sphere space to represent clusters that makes it somewhat generate not very good linearly separable classes.

Hence, to fight off the negative effects of CC we hypothesize that having uniformly distributed clusters in the hyper-sphere is a desire able quality to learn better linearly separable representations. We also hope that such representation would be not be prone to incorrect K, and so will exhibit robustness towards the number of classes K to cluster together data points that end up having similar feature. So we introduce a newer loss that combines the uniformity loss and the contrastive loss with some weight parameter $\lambda \in [0, 1]$ as follows

$$L = \lambda L_{cont} + (1 - \lambda) L_{unif}$$

In this case we can think of L_{cont} as our alignment loss whose job is to compute which point should be close to which point in the representation as it is a mix of L_{inst} and L_{clu} that allow us to cluster the given data points. Similarly L_{unif} can be thought of as imposing uniformity on the hyper-sphere representation i.e the clusters are uniformly distributed on the hyper-sphere and so we can exploit the full hyper-sphere this way.

If we let $\lambda = 0$, we would only be enforcing the uniformity loss without any kind of pairing or clustering in the representation, while letting $\lambda = 1$ would essentially recover the CC model. We did not experiment further with the weighting but one future direction could be trying to weight the L_{inst} and L_{clu} separately within the L_{cont}.

5. Experiments

We put our model to test using different experimental settings. First experiment revolves around the linearly separable representations and utilizing the hyper-sphere fully to its capabilities. For visualization purposes we use $d = 3$ hence the hyper-sphere is just a 3-d sphere in our case, however these results can be expanded without any trouble to higher dimensional spaces. The second experiment tries to empirically verify the effect of training the models on wrong K (number of classes) and using the elbow method on the representations, coupled with k-means to see if we our representation would be able to learn representations that are accurate to the data regardless of the K value.

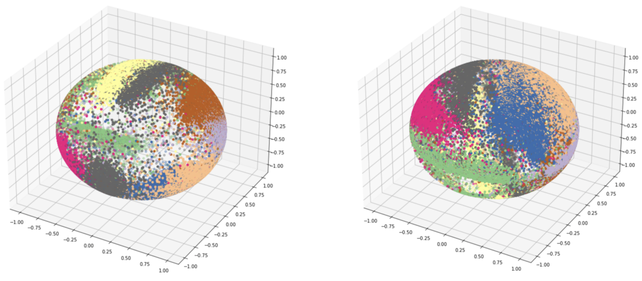

Figure 5. Left : Representation on the unit hyper-sphere generated by the vanilla CC. Right : Representation on the unit hyper-sphere generated by our RCC.

The model architecture, hyper parameter configurations and implementation details are as follows :

- Both CC and RCC use ResNet18 as their backbone architecture to generate 3 dimensional representations. We apply an L2 Norm after the final pass over the backbone to generate vectors that are normalized and are contained within the unit hyper-sphere.

- Both models use 2 layer MLP for their instance and cluster level projections. The instance level projection head utilizes a ReLU as its activation function while the cluster level projection head utilizes Softmax in its final layer.

- We are using $\lambda = 0.95$ for our results and this value is sub-optimal. The temperature parameter for instance contrastive loss was 0.5 while the temperature parameter for cluster contrastive loss was set to 1.0.

- We are using the Adam optimizer with $lr = 3e^{-4}$, weight decay $= 0$, and train for 200 epochs each.

- All the models were trained 5 times and the results were averaged out to make sure the results are not random or affected by seed.

- We train all the models on MNSIT dataset with a batch size of 256, and each image resized to 64x64.

- The family of augmentations we used includes : Gaussian Blur, ResizedCrop, ColorJitter and grayscale augmentations applied randomly to the data points.

The experiment specific settings and findings can be found in their own sub headings below.

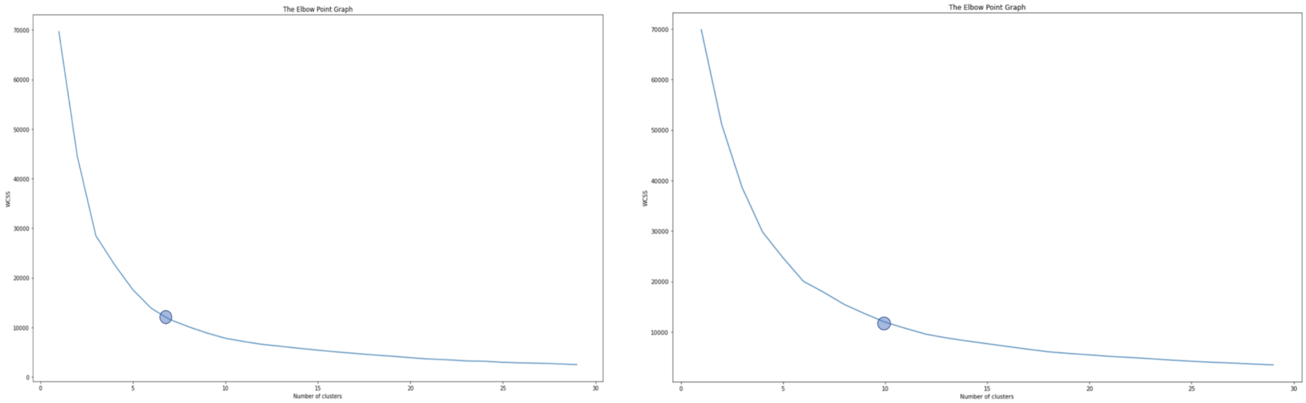

Figure 6. Elbow Method with k-means applied to the two models for $k \in (1, 30)$. Left : Elbow method applied on the representation generated by CC. It shows that the best K value for the representation occurs at approximately K = 6. Right : Elbow method applied on the representation generated by RCC. It shows that the best K value for the representation occurs at approximately K = 10, which was the ground truth.

5.1. Experiment 1 : Utilizing Unit Hyper Sphere

We hypothesize that if we were to enforce some kind of uniformity on our representation in the unit hyper-sphere we would be able to generate linearly separable representation, and this representation would allow us to better utilize and exploit the unit hyper-sphere. We tested this out and the results are discussed below.

- Our clustering accuracy increased with using the uniformity loss on MNIST with $\lambda = 0.95$. We obtained a clustering accuracy of 92% with our RCC compared to 89% on CC after 5 runs.
- The 3-D representations for each of the models are displayed in Figure 5 and it is obvious that the representation generated by the CC results in abundance of space being left out and not being able to exploit the unit hyper-sphere. Contrary to the CC, our method however fully utilizes the empty space.
- To test if adding uniformity results in better linearly separable cluster in representations we decided to design a downstream task which would use our representation to train a linear classifier. We trained a linear classifier for 5 epochs with Adam optimizer with a $lr = 1e^{-4}$. The linear classifier generated an accuracy of 82% on CC while 87% on RCC.

The results of the experiments empirically support our claims that adding uniformity allows us to make use of the unit hyper-sphere to a better extent all the while letting us generate a representation that is better linearly separable.

5.2. Experiment 2 : Robustness to K

We also hypothesized that having a better linearly separable representation in the unit hyper-sphere would allow us to cluster the data in such a way that it is more robust to incorrect value of K. That would be because we believe that having clearly defined linearly separable clusters as representation means that the model is looking at the features of the images to put them together or push them together instead of actually just relying on the K. This would mean that even if our model was to cluster the data into K distinct linearly separable cluster, each cluster should behave like a bi-modal distribution where we can clearly observe the hyper plane that would separate the two different classes within one squeezed class. The method and the results are discussed below :

- We train the 2 models with K = 5 on MNIST dataset where the true K = 10. We do this to see if the model is able to pick up the notion of 10 clearly distinct classes compared to instead of 5 which the model was fed in.
- After training the 2 models we perform k-means on the learned representation as our downstream task with the values of $K \in (1, 30)$. We save the Within Cluster Sum of squared distances (WCSS) for each of the values of K and plot them. Then we visually apply the elbow method to see what would be the best K value our learned representation. The results are showed in Figure 6.

The results of the elbow method coupled with the k-means suggest that the representation we learned from our RCC method seems to be able to grasp the ground truth K value (10) and hence it can be said based on these empirical results that the RCC method does indeed provide robustness to the clustering algorithm. A point to note is that elbow method may not be a very reliable measure in this case but due to the lack of time we were not able to find implement something to go along with it.

6. Conclusion

In this paper we proposed robust contrastive clustering (RCC) method for a robust, online clustering framework for images that is suitable to work with large scale systems, like anomaly detection and recommendation systems, with unlabelled datasets and unknown number of classes as long as there exists a reasonable, educated guess. Our RCC is based upon the Contrastive Clustering method (CC) (Li et al., 2021) using the the ”respresentation as label” approach and 2 projection heads for our contrastive losses. With the added uniformity, we have empirically shown that our method results in better linearly separable representation in the hyper-sphere that is error prone to incorrect guesses to number of classes in the data set making our approach a better alternative to CC for downstream tasks. Going further we would like to work on better metrics of uniformity in hyper-sphere and use our learned representation to measure the performance on some of the more popular downstream tasks in self-supervised learning.

References

Ahmed, M., Mahmood, A. N., and Hu, J. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19, 2006.

Berkhin, P. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pp. 25–71. Springer, 2006.

Bruse, J. L., Zuluaga, M. A., Khushnood, A., McLeod, K., Ntsinjana, H. N., Hsia, T.-Y., Sermesant, M., Pennec, X., Taylor, A. M., and Schievano, S. Detecting clinically meaningful shape clusters in medical image data: metrics analysis for hierarchical clustering applied to healthy and pathological aortic arches. *IEEE Transactions on Biomedical Engineering*, 64(10):2373–2383, 2017.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 132–149, 2018.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Chuang, K.-S., Tzeng, H.-L., Chen, S., Wu, J., and Chen, T.-J. Fuzzy c-means clustering with spatial information for image segmentation. *computerized medical imaging and graphics*, 30(1):9–15, 2006.

Coates, A. and Ng, A. Y. Learning feature representations with k-means. In *Neural networks: Tricks of the trade*, pp. 561–580. Springer, 2012.

Coleman, G. B. and Andrews, H. C. Image segmentation by clustering. *Proceedings of the IEEE*, 67(5):773–785, 1979.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Li, Y., Hu, P., Liu, Z., Peng, D., Zhou, J. T., and Peng, X. Contrastive clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8547–8555, 2021.

Ng, A., Jordan, M., and Weiss, Y. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.

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

Wang, T. and Isola, P. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pp. 9929–9939. PMLR, 2020.