Unsupervised Image Classification of Fish Without the Inference of Cluster Number

Akhilesh Varma Bhupathiraju
Faculty of Computer Science
Dalhousie University
Halifax, Canada
ak445438@dal.ca

Christopher Whidden
Faculty of Computer Science
Dalhousie University
Halifax, Canada
cwhidden@dal.ca

***Abstract*—We aim to implement a deep learning based clustering algorithm (DeepDPM) for clustering fish images into different classes. Unlike other unsupervised methods this is a non-parametric method that does not require specifying of the number of clusters at the beginning as the algorithm infers it during the learning. We have replicated the results of DeepDPM algorithm on the MNIST, Fashion-MNIST, USPS, STL10, and ImageNet datasets as showcased by them. We now plan to implement the algorithm on the Fish4-Knowledge dataset consisting of unlabelled fish images of 23 different species in order to cluster rare fish group. This work will help enable automated monitoring and detection of unknown species of fish around commercial infrastructure.**

***Index Terms*—DeepDPM, upsupervised classification, EM, SCAN, split-merge, feature-extraction**

I. Introduction

Monitoring the movement of fish is essential for number of reasons from maintaining a well balanced marine ecosystem and to estimating the fish stock present in an area [1]. Most of the observation of fish involves a large number of time-consuming manual effort by experts. There has been some promising work done by applying machine learning techniques to detect and classify fish by species. For instance, Knausgård et al has developed a deep learning system [9] to detect temperate fish, while Shafait et al on the other hand has developed a system [11] which identifies and counts fish from videos captured in uncontrolled underwater environment. However, this still requires large amounts of training data labeled by experts. Moreover, with such a wide variety of fish present it is not always possible to keep a track of them without active human involvement. This consumes a lot of time and resource which can be instead put to a better use.

To address this issue unsupervised clustering can be applied which automatically groups similar species of fish together. To monitor the underwater fish, Innovasea has developed a deep learning based system [8] which automatically counts and classifies the fish by combining the video and sonar data. This has been deployed at White Rock hydroelectric dam and is being tested [3]. Despite it being automated, the trick in this is that system needs to be taught to recognize the fish caught on camera as salmon or trout, red fish or blue fish [2]. With the parameterized models, there is some assumption that is made on the data. While, non-parametric methods are a set of algorithms that do not make any prior assumptions about the mapping function of training data. In case of clustering, the value of K (approximate number of clusters) should be provided by the user, and there is always a question of what is the right value of K. If the K value is not approximated correctly, it may not always result in a optimal number of clusters. One way to deal with this is running the clustering algorithm multiple times with different set of K values and choosing the one which gives the best results. However, this has several drawbacks which makes it difficult to apply in practice. The primary one being, it is not feasible to run the algorithm multiple times on large datasets particularly in Deep Learning as it not feasible to train with multiple different number of clusters it is computationally expensive to train and consumes a lot of energy which in turn causes a negative impact on the environment, moreover it cannot be scaled . Here is where non-parametric algorithms like DeepDPM [10] proposed by Ronen et al come into play, being a non-parametric model it does not require for the user to mention the K value, the model estimates it by changing the K value using split-merges of clusters. It also makes use of a novel inference for Expectation-Maximization (EM). This algorithm is ideal for our use-case as it scales to large datasets and it can also handle imbalance in dataset. In this paper, we plan to implement the DeepDPM algorithm and test it on the real-world fish dataset to see how well the model would be able to identify different classes of unlabelled fish without much manual intervention.

II. Methods

DeepDPM algorithm can be looked as a DPM (Dirichlet Process of Mixture) inference algorithm. The mixture model is a GMM (Gaussian Mixture Model) and can be perceived containing infinitely-many Gaussians.

$$p\left(\boldsymbol{x} \mid (\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)_{k=1}^{\infty}\right) = \sum_{k=1}^{\infty} \pi_k \mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \tag{1}$$

where $\mathcal{N}\left(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$ is a Gaussian probability density function (pdf) (of mean $\mu_k \in \mathbb{R}^d$ and a d-by- d covariance ma-

Mitacs Accelerate

trix Σ_k) evaluated at $x \in \mathbb{R}^d, \pi_k > 0 \forall k$, and $\sum_{k=1}^{\infty} \pi_k = 1$. Let $\boldsymbol{\theta}_k = (\boldsymbol{\mu}_k, \Sigma_k)$ denote the parameters of Gaussian k. Note the distinction between component k (namely, the k-th Gaussian, identified with its parameter, θ_k) and cluster k. The components, $\boldsymbol{\theta} = (\boldsymbol{\theta}_k)_{k=1}^{\infty}$, and weights, $\pi = (\pi_k)_{k=1}^{\infty}$, are assumed to be drawn (independently) from their own prior distributions: the weights, π, are drawn using the Griffiths-Engen-McCloskey stick-breaking process (GEM) [51] with a concentration parameter $\alpha > 0$, while the parameters, $(\theta_k)_{k=1}^{\infty}$, are independent and identically-distributed (i.i.d.) draws from their prior $p(\boldsymbol{\theta}_k)$, typically a Normal-Inverse Wishart (NIW) distribution. While there are infinitely-many components, there are still finitely-many clusters as the latent random variable K is bounded above by N. The DeepDPM algorithm primarly contains two steps which feature extraction and clustering.

In feature extraction there are two ways of doing it: an end-to-end approach in which the features and clustering are learned simultaneously, and a two-step approach in which clustering is performed on pre-computed latent features. For the later method, feature extraction of SCAN [12] is used.

The clustering part of algorithm uses split-merge as inspired [3] to change the K value where cluster has a subcluster pair associated with it. It has two primary parts, the first is clustering net, and the second contains K subclustering nets (one for each cluster k). During training split-merge is used to change the K value as in [3]. As K changes, the model architecture also changes including the last layer of the clustering net. A new loss is used here by the EM (Expectation Maximization) in the Bayesian GMM. Due to the new amortized EM the prediction of points improve not only in the current batch but also in the other batches. The smoothness of the function caused by this also serves as an inductive bias such that points which are close in the observation space should have similar labels.

III. Results

The experiments were conducted on both locally on 2.5 GHz Core Intel i5 7th Gen with 12GB of RAM and running Windows 10 operating system as well as on the Deep Sense platform having a 2xp100 GPU with 20 Core IBM Power8NVL 4.0GHz and running Redhat Enterprise 7.7 operating system. Table 1 shows the results that were reproduced on both balanced and imbalanced datasets of MNIST [6], Fashion-MNIST [13], USPS [7], STL10 [4], and ImageNet [5].

The evaluations is done based on three common metrics: clustering accuracy (ACC) which is defined as

$$\mathrm{ACC} = \max_m \left(\frac{\sum_{i=1}^{N} \mathbb{1}\left(y_i = m\left(z_i\right)\right)}{N} \right) \tag{2}$$

where N is the number of data points, y_i is the Ground-Truth (GT) class label of data point i, z_i is the predicted cluster assignment according to the clustering algorithm under consideration, $\mathbb{1}(\cdot)$ is the indicator function, and m is defined by all possible one-to-one mappings between the predicted class membership and the ground-truth one.

; Normalized Mutual Information (NMI) which is defined by

$$\mathrm{NMI} = \frac{2 \times I(\boldsymbol{y}; \boldsymbol{z})}{H(\boldsymbol{y}) + H(\boldsymbol{z})} \tag{3}$$

where $H(.)$ stands for entropy and $I(.;.)$ denotes Mutual Information (MI). One problem with this metric, however, is that the MI term, which appears in the numerator, does not penalize large cardinalities (i.e., over clustering). NMI is not sensitive enough to over clustering. ; Adjusted Rand Index (ARI) The Rand index (RI) quantifies the percentage of "correct" decisions for each pair of data points. A decision is correct if two examples either belong to the same GT class and the same cluster assignment (a true positive, TP), or being from two different GT classes and assigned to two different clusters (a true negative, TN). Similarly, clustering errors are false positives (FP) and false negatives (FN). Then RI is computed by:

$$RI = \frac{TP + TN}{TP + TN + FP + FN}. \tag{4}$$

The ARI measure is the corrected-for-chance version of the Rand index. Given a set $\mathcal{S}$ of N elements, and two groupings or partitions (e.g. y and z) of these elements, the overlap between y and z can be summarized in a contingency table $[c_{kl}]$ where each entry c_{kl} denotes the number of objects in common between y_k and z_k : $c_{kl} = |y_k \cap z_k|$. Let a_k be the sum (f each row, meaning, $a_k = \sum_l c_{kl}$, and b_k the sum of each column, i.e. $b_k = \sum_k c_{kl}$. The ARI measure is then calculated by:

$$ARI = \frac{\sum_{kl} \binom{n_{kl}}{2} - \left[\sum_k \binom{a_k}{2} \sum_l \binom{b_l}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_k \binom{a_k}{2} + \sum_l \binom{b_l}{2}\right] - \left[\sum_k \binom{a_k}{2} \sum_l \binom{b_l}{2}\right] / \binom{n}{2}}$$

The higher values in them the better. (5)

From Table 1, we see that the algorithm was able to infer the value of K fairly well for all the datasets. There were cases when it wasn't able to get the exact K value for example in USPS and ImageNet but it was right in the ballpark zone as conducted in the previous experiments. The model underestimated the K value in the case of USPS dataset while of ImageNet dataset the K value was overestimated. If we train the model longer time with more number of epochs for USPS dataset, the model might have arrived at the right value of K. The accuracy score along with the other evaluation metrics meet the expectations.

Datasets Used	Evaluation Metrics				
	NMI	ARI	acc	Paperacc	K
MNIST	0.941	0.953	0.9787	0.98	10
MNIST-imbalanced	0.941	0.953	0.978	0.98	10
Fashion-MNIST	0.687	0.534	0.663	0.62	10
Fashion-MNIST-imb	0.670	0.515	0.612	0.61	10
USPS	0.826	0.70137	0.727	0.89	7
USPSimbalanced	0.864	0.807	0.809	0.94	8
STL10	0.768	0.670	0.832	0.85	10
ImageNet	0.736	0.515	0.645	0.66	52

Table 1. Results obtained from replicating the DeepDPM algorithm on different datasets as mentioned by the author.

Datasets Used	Train samples	Val samples	Data dimension	GT K
MNIST [6]	60,000	10,000	28 $\times$28	10
USPS [10]	7291	2007	16 $\times$16	10
Fashion-MNIST [13]	60,000	10,000	28 $\times$28	10
STL10 [4]	5,000	8,000	96 $\times$96 $\times$ 3	10
ImageNet-50 [5]	64,274	2,500	224 $\times$224 $\times$ 3	50

Table 2. Description of the datasets used

CONCLUSION

In this paper we presented the results conducted using DeepDPM on MNIST, Fashion-MNIST, USPS, STL10, and ImageNet datasets on both balanced and imbalanced to cluster the data without the prior knowledge of the number of clusters. We also created a script to view the cluster outputs by mapping them back to their original class labels. This showed all the misclassified images by the model in each cluster. Executing the code was a challenge in itself because when we run the clustering script locally without any GPU it would typically take 6-8 hours to run with pretrained embeddings even on a relatively small dataset like MNIST. So, we made use of the DeepSense servers to run the model but again it had package dependency issue initially. The end results were quite promising as the model was successfully able to identify the exact number of clusters most of the times without prior mentioning of it. And it performed equally well for both balanced and unbalanced datasets.

FUTURE WORK

DeepDPM has so far been tested only on the standard datasets, but not real datasets of fish. The next step is to test the algorithm on a dataset of real fish. We will be writing a visualization code to visualize and analyze the results of the clustering in order to verify the results and identify opportunities to improve the clustering method. We will be running the algorithm on Fish4Knowledge dataset which consists of 27,370 images of 23 species of fish captured from a live video data. For this, we will be adapting the methods used to train and test DeepDPM on ImageNet, including the feature extraction and clustering steps. Finally, we will apply the model in conjunction with partners at Innovasea to enable unsupervised clustering at the White Rock Dam test site in Nova Scotia, Canada. The clusters identified by DeepDPM will be analyzed by fish experts to rapidly create a set of training data used to train a low power YOLO model for continuous detection and identification of fish. This work will help enable rapid scaling of machine learning models for detecting and classifying fish around commercial infrastructure by greatly reducing the work required to identify and label fish species.

ACKNOWLEDGEMENT

This project is being done in collaboration with InnovaSea Marine System Canada and it supported by the MITACS Accelerate Program.

REFERENCES

[1] 2022. [online]. available: https://nlai.blue/the-importance-of-monitoring-marine-ecosystems-for-sustainable-ocean-resource-use/. [accessed: 08-jul- 2022]., 2022.

[2] B. dean-simmons, "innovasea testing a faster way to count fish in nova scotia with artificial intelligence — saltwire", saltwire.com, 2022. [online]. available: https://www.saltwire.com/atlantic-canada/business/innovasea-testing-a-faster-way-to-count-fish-in-nova-scotia-with-artificial-intelligence-100732253/. [accessed: 08- jul- 2022]., 2022.

[3] Jason Chang and John W Fisher III. Parallel sampling of dp mixture models using sub-cluster splits. *Advances in Neural Information Processing Systems*, 26, 2013.

[4] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[6] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[7] J. J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

[8] Vishnu Kandimalla, Matt Richard, Frank Smith, Jean Quirion, Luis Torgo, and Chris Whidden. Automated detection, classification and counting of fish in fish passages with deep learning. *Frontiers in Marine Science*, page 2049, 2022.

[9] Kristian Muri Knausgård, Arne Wiklund, Tonje Knutsen Sørdalen, Kim Tallaksen Halvorsen, Alf Ring Kleiven, Lei Jiao, and Morten Goodwin. Temperate fish detection and classification: a deep learning based approach. *Applied Intelligence*, 52(6):6988–7001, 2022.

[10] Meitar Ronen, Shahaf E Finder, and Oren Freifeld. Deepdpm: Deep clustering with an unknown number of clusters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9861–9870, 2022.

[11] Faisal Shafait, Ajmal Mian, Mark Shortis, Bernard Ghanem, Phil F Culverhouse, Duane Edgington, Danelle Cline, Mehdi Ravanbakhsh, James Seager, and Euan S Harvey. Fish identification from videos captured in uncontrolled underwater environments. *ICES Journal of Marine Science*, 73(10):2737–2746, 2016.

[12] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer, 2020.

[13] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017. cite arxiv:1708.07747Comment: Dataset is freely available at https://github.com/zalandoresearch/fashion-mnist Benchmark is available at http://fashion-mnist.s3-website.eu-central-1.amazonaws.com/.