Convex Optimization for Active Learning

Aditya Jain
Institute for Aerospace Studies
University of Toronto
Toronto, Canada
aadi.jain@mail.utoronto.ca

Safwan Jamal
Dept. of Electrical and Computer Engineering
University of Toronto
Toronto, Canada
safwan.jamal@mail.utoronto.ca

***Abstract*—Insects play a fundamental role in biodiversity, ecosystem health, and ecological services such as pollination, yet few species are monitored adequately. Camera traps coupled with machine learning (ML) models for automatic species identification can massively scale insect monitoring efforts. The limiting factor remains around the need to label millions of camera trap images for model training. *Representatives* or *exemplars*, are a subset of data points that well encode the entire dataset. This small subset of data is tractable for annotation by the experts and can be used to train ML models, without much compromise on the model accuracy. We frame this task of finding exemplars as a convex optimization problem. We develop the theoretical formulation and quantitatively evaluate the effectiveness of the framework on three image classification datasets.**

I. Introduction

Climate change and other anthropogenic factors are having an unprecedented impact on the earth's biodiversity [1]. With nearly 1 million described species [16], insects are the bulk of all described species on the planet, yet few species are monitored adequately. The taxa of special interest are moths, which represent nearly one-fifth of all insect species and are foundational in their roles as pollinators and as sustenance for birds and other organisms. Camera traps for moth monitoring [2] have started to gain interest over traditional methods [6], [15]. The amount of image data collected by these traps can amount to hundreds of thousands on a few nights of operation. While [2], [8] propose machine learning and computer vision techniques to automatically process this moth trap data, their approaches involve manual annotation of insect images to train a fine-grained image classifier [17]. It is challenging to find domain experts for annotation and it is time-consuming to label such camera trap data.

A subset of data points, called *representatives* or *exemplars*, which can efficiently describe the entire dataset, will significantly reduce labeling effort, compute requirements for model training, and allow the use of smaller models which can be directly deployed on the camera traps. Active learning, a subfield of machine learning, is an area of research that attempts to solve this problem [13], [14]. It is the method of progressively selecting and annotating the most informative unlabeled samples, to obtain a high classification performance. Several query strategy frameworks exist in the literature such as uncertainty sampling and expected model change [14].

The active learning problem in the framework of convex optimization has been proposed by [4], [5]. In [4], the authors find representative examples from the available dataset, by minimizing the joint cost of representation encoding and the number of representative examples chosen. The exemplars will then be annotated for training or fine-tuning the model. In [5], they additionally incorporate the model prediction confidence in the minimization objective. This work reproduces and validates [4]. The proposed framework is a row-sparsity regularized trace minimization program whose objective is to find a *few representatives* that *well encode* the entire dataset, given dissimilarities between the data points. The regularization parameter can be tuned to sample a specific amount of data from the entire dataset. To evaluate the representation quality of the exemplars given by the optimization program, we train and evaluate a custom convolutional neural network (CNN) [10] on three image classification datasets. The CNN is trained on the exemplars and its accuracy is compared with the same CNN models trained on randomly selected points and the full dataset. We demonstrate that training on exemplar data improves the model accuracy as compared to training on randomly selected points and training on all of the data points.

II. Problem Statement

We consider the problem of finding representatives from a collection of N data points. Assume we are given a set of non-negative dissimilarity values d_{ij}, $\forall i, j = 1, ..., N$, where d_{ij} corresponds to the dissimilarity between data point i and data point j. If they are not given, the dissimilarities can be calculated using Euclidean distances or the inner product between the data points. A smaller d_{ij} indicates that data point i is a good representative of data point j. The dissimilarity values can be arranged in a matrix form

$$D \triangleq \begin{bmatrix} d_1^T \\ \vdots \\ d_N^T \end{bmatrix} = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,N} \\ \vdots & \vdots & & \vdots \\ d_{N,1} & d_{N,2} & \cdots & d_{N,N} \end{bmatrix} \in \mathbb{R}^{N \times N}, \quad (1)$$

The goal is to select a few data points that best represents the entire dataset using the dissimilarities D. We now define a probability matrix of the form

$$Z \triangleq \begin{bmatrix} z_1^T \\ \vdots \\ z_N^T \end{bmatrix} = \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,N} \\ \vdots & \vdots & & \vdots \\ z_{N,1} & z_{N,2} & \cdots & z_{N,N} \end{bmatrix} \in \mathbb{R}^{N \times N}, \quad (2)$$

where z_{ij} represents the probability of data point i being a representative of data point j, hence $z_{ij} \in [0, 1]$. A data point j can have multiple representatives, in which case $z_{ij} > 0$ for all of the corresponding i indices of the representatives. Therefore, we must have $\sum_{i=1}^{N} z_{ij} = 1, \forall j$.

Since the goal is to select a few representatives that well encode the entire dataset, our proposed optimization problem aims to minimize an objective function consisting of two terms. The first is the total encoding cost of the dataset with the representatives. If the data point i is chosen as the representative of data point j with probability z_{ij}, the cost of encoding j with i is $d_{ij}z_{ij} \in [0, d_{ij}]$. Thus, the encoding cost of j using all of its representatives is $\sum_{i=1}^{N} d_{ij}z_{ij}$ and the encoding cost for the entire dataset is $\sum_{j=1}^{N}\sum_{i=1}^{N} d_{ij}z_{ij}$. The second term in the objective function is the cost associated with the number of representatives. It is established earlier that if sample i is a representative of another sample j, then $z_{ij} \neq 0$, i.e., the row z_i is non-zero. Therefore, the number of non-zero rows in Z effectively determines the number of representatives. Adding the two terms gives the following optimization problem

$$\begin{aligned} &\min \sum_{j=1}^{N}\sum_{i=1}^{N} d_{ij}z_{ij} + \lambda \sum_{i=1}^{N} \mathbb{I}(\|z_i^T\|_q) \\ &\text{s.t. } z_{ij} \geq 0,\ \forall i, j;\ \sum_{i=1}^{N} z_{ij} = 1, \forall j, \end{aligned} \tag{3}$$

where $\mathbb{I}$ is an indicator function, which outputs the value zero if its argument is zero and one otherwise. The first term in the objective function represents the total cost of encoding all data points using the representatives and the second term corresponds to the cost associated with the number of representatives. The regularization parameter $\lambda \geq 0$ controls the trade-off between the two terms. However, counting the number of non-zero rows of Z is, in general, NP-hard and therefore, we must perform a convex relaxation in the objective function. The problem thus becomes

$$\begin{aligned} &\min \sum_{j=1}^{N}\sum_{i=1}^{N} d_{ij}z_{ij} + \lambda \sum_{i=1}^{N} \|z_i\|_q \\ &\text{s.t. } z_{ij} \geq 0,\ \forall i, j;\ \sum_{i=1}^{N} z_{ij} = 1, \forall j, \end{aligned} \tag{4}$$

where instead of counting the number of non-zero rows of Z, we use the sum of the ℓ_q-norms of the rows of Z. For (4) to be a convex problem, we can choose $q \in \{2, \infty\}$. The optimization problem (4) can also be expressed in matrix form as

$$\min \text{tr}(D^T Z) + \lambda \|Z\|_{1,q} \ \text{ s.t. } Z \geq 0,\ \mathbb{1}^T Z = \mathbb{1}^T, \tag{5}$$

where $\text{tr}(\cdot)$ is the trace operator, $\lambda\|Z\|_{1,q} \triangleq \lambda \sum_{i=1}^{N} \|z_i\|_q$, and $\mathbb{1}$ represents a vector of length N which has all elements equal to one. We can sweep through the value of λ to choose the number of representatives the optimization program finds. For small values of λ, the emphasis is on better encoding of the data points via the representatives. As $\lambda \to 0$, all points are selected as representatives because a point can be best represented by itself, i.e., $z_{ii} = 1\ \forall i$. For larger values of λ, there is more emphasis on the row-sparsity of Z, which gives us a smaller number of representatives. As $\lambda \to \lambda_{max}$, we select only one representative for all of the data points. This is displayed in Figure 1. The value of λ_{max} depends on the choice of ℓ_q-norm. We choose to work with ℓ_∞-norm and λ_{max} is calculated as

$$\lambda_{max,\infty} \triangleq \max_{i \neq \ell} \frac{\|d_i - d_\ell\|_1}{2}, \tag{6}$$

where d_i is the i^{th} row of dissimilarity matrix D (1). The theoretical derivation for (6) can be read in [4]. Once the optimization program (5) is solved, the representative examples are the indices of non-zero rows of Z.

III. Experiments

In this section, we discuss the methodology and experiments for the quantitative evaluation of the representative data points from the optimization program. Experiments are conducted on three image classification datasets. For each of the datasets, we randomly sample 20%, 50%, and 80% of the data points from the training set. For the same percentages, we also find the exemplar data points for each dataset using the optimization program. Multiple copies of the same CNN model are trained on the random, exemplar, and full training sets. The model accuracy on the test set is then compared across all of the runs and displayed in Figure 2. The details are discussed below.

Datasets: The three image classification datasets are: CIFAR-10 [9], Oxford-IIIT Pet [12], and Butterfly & Moths [7]. CIFAR-10 is one of the earliest image classification benchmark datasets, consisting of 6000 32x32 RGB images in 10 classes, with 6000 images per class. The 10 object categories consist of animals and vehicles. The Oxford-IIIT Pet is a dataset consisting of 37 classes of dogs and cats, with roughly 200 RGB images for each class. The Butterfly & Moths is a dataset of 100 species of butterflies and moths, consisting of roughly 120 training images per species. We call this dataset Lepidoptera, the order of insects comprising butterfly and moth species. In order to have a more tractable optimization program, we work with reduced versions of the datasets. For CIFAR-10, we create the training and testing sets by randomly sampling 150 and 40 images per class, respectively. For Oxford-IIIT Pet, we randomly select 10 classes out of the 37 and randomly divide the data into 150 and 40 images for training and testing respectively. Similarly for the Lepidoptera dataset, we randomly choose 10 out of the 100 species and select 100 and 5 images for training and testing, respectively.

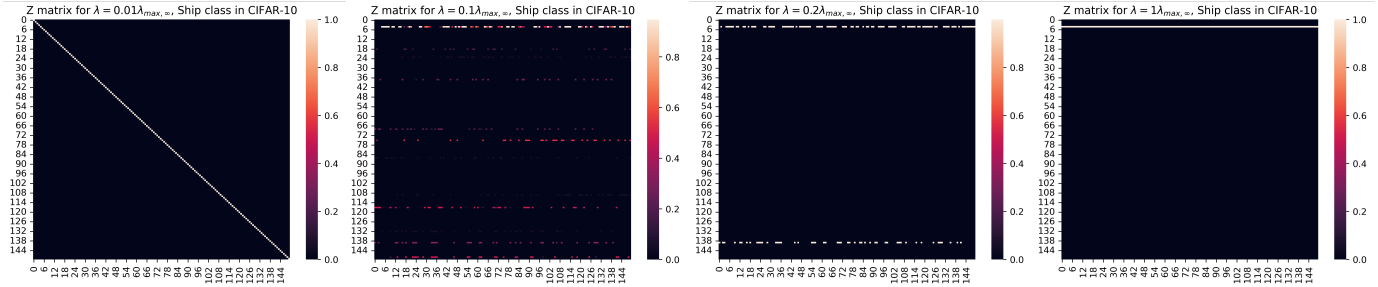

Fig. 1. Matrix Z obtained after solving the proposed optimization program (5) for the ship class in CIFAR-10. The plots are for several values of λ, where $\lambda_{max,\infty}$ is defined in (6).

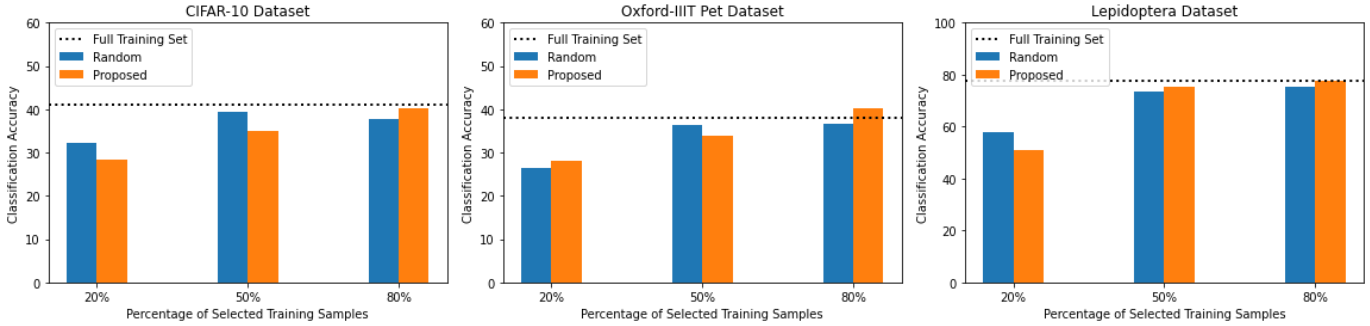

Fig. 2. Test classification accuracies (%) on the CIFAR-10 (left), Oxford-IIIT Pet (middle), and Lepidoptera (right) datasets. The horizontal axis shows the percentage of the selected representatives from each class. The dashed line shows the classification accuracy using all of the training samples.

Dissimilarity matrix: For a given dataset, the D matrix (1) is formed for each class using all of the training images. Every image is downsampled to 224×224 (except for CIFAR-10 images) and converted into grayscale. Each image is now represented by an n-dimensional vector obtained by vectorizing its grayscale form, where n is 1024 for CIFAR-10 and 50176 for the rest. The dissimilarity d_{ij} in D is the Euclidean distance between the vectorized versions of image i and image j.

Exemplars: We now find the exemplar data points by solving the convex program (5) for each class in a dataset. The regularization parameter λ is tuned to sample a particular number of representatives for a class. As discussed in section II, the representative examples are the indices of non-zero rows of Z (1) after solving (5). Figure 1 shows the matrix Z for the ship class in CIFAR-10 for several values of λ. We collate the exemplar points for every class in the dataset to form the exemplar training set.

Model architecture: We use the same CNN architecture for all of the training instances. The input to the neural network is a 32×32×3 image for CIFAR-10 and 224×224×3 for others. The first hidden layer convolves 6 5×5 filters with stride 1 with the input image, followed by a layer of rectified nonlinearity [11] and max-pooling. The second hidden layer is similar but convolves 16 5×5 filters. This is followed by two fully-connected layers consisting of 120 and 84 rectifier units respectively. The output layer is a fully-connected linear layer with a single output for each class in the dataset.

Training: For each dataset, the CNN model is trained seven times: thrice for the random subset, thrice for the exemplar subset, and once on the full training set. The same set of hyperparameters (optimizer, batch size, loss function, and early stopping criteria) are used across all training instances and the model accuracy is evaluated against the same test set.

Figure 2 shows the training results. When training on a small subset of data (e.g. 20%), there is no clear benefit in using the exemplar points. However, when increasing the size of the subset, the exemplar model accuracy is consistently higher than the random set. And for 80%, the exemplar model accuracy is equal to or greater than the full training set model accuracy.

IV. Conclusion

In this work, we frame the task of finding representative examples from a collection of data points as a convex optimization problem. We show through quantitative evaluation on three image classification datasets that the exemplar points found by the optimization program are indeed a good representation of the entire dataset. We see two possible extensions to this work. First, it can be extended to [5] to include the model's prediction confidence in choosing the representatives as part of the optimization program (4). Second, we can extract features in latent space for an image using a model pre-trained on ImageNet [3]. This might result in better representative data points as calculating dissimilarities from Euclidean distances of vectorized images is not invariant to the scaling and rotation of images.

REFERENCES

[1] Céline Bellard, Cleo Bertelsmeier, Paul Leadley, Wilfried Thuiller, and Franck Courchamp. Impacts of climate change on the future of biodiversity. *Ecology letters*, 15(4):365–377, 2012.

[2] K Bjerge, JB Nielsen, MV Sepstrup, F Helsing-Nielsen, and TT Høye. A light trap and computer vision system to detect and classify live moths (lepidoptera) using tracking and deep learning. *bioRxiv*, 2020.

[3] 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. Ieee, 2009.

[4] Ehsan Elhamifar, Guillermo Sapiro, and Rene Vidal. Finding exemplars from pairwise dissimilarities via simultaneous sparse recovery. *Advances in Neural Information Processing Systems*, 25, 2012.

[5] Ehsan Elhamifar, Guillermo Sapiro, Allen Yang, and S Shankar Sasrty. A convex optimization framework for active learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 209–216, 2013.

[6] D Groenendijk and WN Ellis. The state of the dutch larger moth fauna. *Journal of insect conservation*, 15(1):95–101, 2011.

[7] kaggle.com. Butterfly & Moths Image Classification 100 species. 2022. Available from: https://www.kaggle.com/datasets/gpiosenka/butterfly-images40-species [30 November 2022].

[8] Dimitri Korsch, Paul Bodesheim, and Joachim Denzler. Deep learning pipeline for automated visual moth monitoring: insect localization and species classification. *INFORMATIK 2021*, 2021.

[9] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[10] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE, 2010.

[11] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[12] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[13] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM computing surveys (CSUR)*, 54(9):1–40, 2021.

[14] Burr Settles. Active learning literature survey. 2009.

[15] Robin Steenweg, Mark Hebblewhite, Roland Kays, Jorge Ahumada, Jason T Fisher, Cole Burton, Susan E Townsend, Chris Carbone, J Marcus Rowcliffe, Jesse Whittington, et al. Scaling-up camera traps: Monitoring the planet's biodiversity with networks of remote sensors. *Frontiers in Ecology and the Environment*, 15(1):26–34, 2017.

[16] Nigel E Stork et al. How many species of insects and other terrestrial arthropods are there on earth. *Annual review of entomology*, 63(1):31–45, 2018.

[17] Yafei Wang and Zepeng Wang. A survey of recent work on fine-grained image classification techniques. *Journal of Visual Communication and Image Representation*, 59:210–214, 2019.