Long-tail learning with attributes

Dvir Samuel[1] Yuval Atzmon[2] Gal Chechik[1,2]
[1]Bar-Ilan University, [2]NVIDIA Research

Abstract

Learning to classify images with unbalanced class distributions is challenged by two effects: It is hard to learn tail classes that have few samples, and it is hard to adapt a single model to both richly-sampled and poorly-sampled classes. To address few-shot learning of tail classes, it is useful to fuse additional information in the form of semantic attributes and classify based on multi-modal information. Unfortunately, as we show below, unbalanced data leads to a "familiarity bias", where classifiers favor sample-rich classes. This bias and lack of calibrated predictions make it hard to fuse correctly information from multiple modalities like vision and attributes.

Here we describe DRAGON, *a novel modular architecture for long-tail learning designed to address these biases and fuse multi-modal information in face of unbalanced data. Our architecture has three classifiers: a vision expert, a semantic attribute expert and a debias-and-fuse module to combine their predictions. We present the first benchmark for long-tail learning with attributes. Using it to evaluate* DRAGON, *we find that it outperforms state-of-the-art long-tail learning models and models of Generalized Few-Shot-Learning-with-attributes (GFSL-a).* DRAGON *also obtains SoTA in some existing benchmarks for single-modality GFSL.*

1. Introduction

Learning with unbalanced data is challenging yet widespread [4, 2, 8, 5]. In most fields and domains, the distribution of samples across classes has a smooth long-tail shape. Such long-tail data poses two major challenges to learning: *data paucity* and *data imbalance*. First, at the tail of the distribution, classes are poorly sampled, and one has to use few-shot and zero-shot learning techniques. The second issue is *data imbalance*. When training a single model for both richly-sampled classes and poorly-sampled classes, models over-represent the rich classes in terms of model parameters. To put it simply, common classes dominate the loss during optimization, and as a result, models devote more of their representation capacity to the rich-sampled classes. Learning with unbalanced data poses a fundamental algorithmic problem because classical learning theory asserts that model capacity should grow with the number of samples.

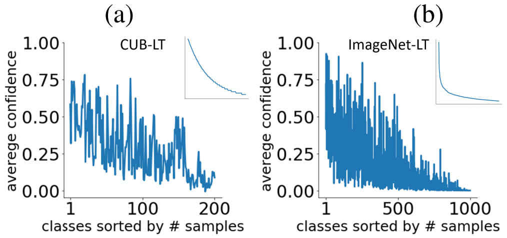

Figure 1: Unbalanced training data leads to a "familiarity bias", where models output more confident predictions for common classes compared with rare ones. Showing average per-class confidence for (a) long-tailed CUB and (b) long-tailed ImageNet. Top right thumbnails show class distributions.

To address *data paucity* for classes at the tail, one can supplement the visual examples with prior information about classes. This information can be given as text or attributes. This approach, *learning with attributes* has been studied mostly for zero-shot and generalized zero-shot learning. Here we propose to adapt it to few-shot learning by fusing information from two modalities. A visual classifier that is trained on visual samples and designed to classify correctly the head classes; and an attribute classifier that is trained on per-class attributes and is expected to operate better on tail classes where the number of samples is small.

Our approach to address *data imbalance* is based on the following observation: Learning with unbalanced data leads to a **familiarity effect**, where models become biased to favor the more familiar, rich-sampled classes. Figure 1 illustrates this phenomenon. Panel 1(a) shows the mean prediction score for each class of a ResNet-101 trained on long-tail variant of the CUB dataset [14]. Namely, for each class, we computed the output of softmax over images from that class. The model outputs more confident predictions when presented with common classes than rare ones. Panel 1(b) shows our preliminary analysis for long-tail Imagenet dataset [9]. It demonstrates that the familiarity bias is also evident in this large-scale dataset.

Interestingly, studies of human decision making and preference learning show a similar bias towards familiar classes. This effect is reported very widely and is related to the availability heuristic studied by Tversky and Kahneman [13].

The familiarity effect caused by data imbalance has a crippling effect on aggregating predictions from multiple modalities. When per-modality predictions are consistently biased, it is difficult to combine information correctly from multiple modalities.

The current paper addresses the *data-imbalance* and the *data paucity* learning challenges, by addressing the familiarity effect in learning with attributes. We describe an easy-to-implement debiasing module that offsets the familiarity effect during training. It further learns to balance information from visual and text modalities. This module has a small number of parameters and is trained end-to-end jointly with the model. Interestingly, it explicitly learns how the number of samples non-linearly contributes to the familiarity bias.

2. Long-tail learning with attributes

Long-tail learning with attributes is normally defined as follows. A training set $\mathcal{X}, \mathcal{Y}$ has n labeled image samples: $\mathcal{X}, \mathcal{Y} = \{(\mathbf{x}_i, y_i), i = 1 \dots n\}$, where each $\mathbf{x}_i$ is a feature vector and $y_i \in \mathcal{Y}$ is a label from $\mathcal{Y} = \{1, 2, \dots k\}$. The samples are drawn from a distribution $\mathcal{D}$ such that the marginal distribution over the classes $p(y)$ is strongly non uniform. For example, $p(y)$ may be exponential $p(y) \sim \exp(-ky)$.

As a second supervision signal, each class y is also accompanied with a *class-description* vector $\mathbf{a}_y$ in the form of semantic attributes or natural-language embedding. For example, in the CUB dataset, a class may be annotated by attributes like `Head-color:red`.

At test time, a new set of samples $\mathcal{X}' = \{\mathbf{x}_i, i = n + 1, \dots, n + m\}$ is given from the same distribution $\mathcal{D}$, where m is the number of samples in the test set. Our goal is to predict the correct class of each test sample.

Our approach: We aim to address the key challenge of long-tail learning from multiple modalities: training a single model to perform well for both head classes and tail classes while fusing information from vision and language.

Our approach is based on two observations: (1) Attribute-based models provide relatively accurate predictions in the low-shot regime. This is because a class can be recognized by its attributes even when few or no training samples are available. (2) The average prediction confidence for samples of a given class is correlated with the number of training samples for that class (Figure 1).

We design an architecture that leverages these observations by learning to adaptively (1) Combine predictions of two experts classifiers: A conventional *visual expert* which is more accurate for head classes and an attribute expert which excels at the tail classes; (2) Reweigh the scores of each class prediction based on its number of training samples. This approach can be viewed as achieving two tasks: It learns (1) to debias the predictions of each model; and (2) to combine the predictions of the two experts.

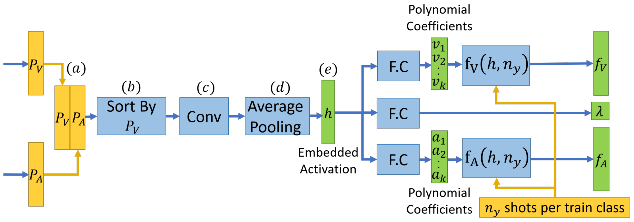

Figure 2: Architecture of the fusion-module for long-tail learning with attributes. The inputs P_V denote the softmax prediction vector of the *Visual Expert*, and and P_A that of the *Attribute Expert*. The outputs f_V, f_A and λ are used in Eq. (1) for re-weighting the inputs as detailed in Section 2.2.

2.1. The architecture

We now describe DRAGON, an architecture for multi-modal long-tail learning. Its design follows two considerations: the model is modular and has few parameters.

Our general architecture takes a late-fusion approach. It consists of two experts modules: A visual expert, which is an image classifier that is trained in a standard way, and an attribute expert that predicts a class-based on a semantic embedding of an image. Each expert outputs a prediction vector which is fed to a fusion-module. The fusion-module takes the experts predictions and learns to debias the familiarity effect, by weighing the experts and rescaling their class predictions.

More formally, given an input image, the architecture evaluates the following score per class y:

$$S(y) \quad = \quad \lambda f_V(y) p_V(y) + (1 - \lambda) f_A(y) p_A(y). \quad (1)$$

Here, $p_V(y)$ and $p_A(y)$ are raw confidence scores of a visual expert and the attribute expert; $f_V(y)$ and $f_A(y)$ are positive coefficients that re-scale the raw confidence scores of the experts to compensate for the familiarity effect; and $\lambda \in (0, 1)$, is a combination coefficient that balances the experts. It learns to exploit the familiarity effect in order to account the more relevant expert.

2.2. A fusion-module

Given an image sample, the fusion-module outputs three components f_V, f_A and λ. $f_V(y)$ is a set of coefficients that reweigh each y class prediction scores for the visual expert and $f_A(y)$ is the same for the attribute expert. The two reweighed expert predictions are combined with a coefficient λ. We leverage the observation that prediction confidence of a class is correlated with its number of training samples (Figure 1) and use the softmax output vectors of the two experts as the input of the fusion-module.

Figure 2 describes the architecture of the fusion-module. It has two main parts. The first part maps the prediction scores to a meaningful embedded space, in four steps. (a) Stack together the predictions of two experts to a $\mathcal{Y} \times 2$ vector. This makes the convolution meaningful across the 2 experts axis. (b) To make it also meaningful along the classes axis, and since classes are categorical, we reorder the classes, by their prediction score according to one of the experts. This sorts the predictions in a meaningful order, and couples together classes that are visually similar that the classifiers tend to confuse. (c) Coupling similar classes allows us to feed the sorted scores to a $N_{filters} \times 2 \times 2$ convolutional network. (d) The convolutional network is followed by an average pooling layer yielding a $(\mathcal{Y}-1) \times 1$ embedding tensor. (e) We denote by h the embedding function described above, and by $\mathbf{h}$ its embedded activation: $\mathbf{h} = h(\mathbf{p}_V, \mathbf{p}_A)$, where $\mathbf{p}_V$, $\mathbf{p}_A$ are vector notations for the prediction scores of each expert.

We now describe the second part, which takes $\mathbf{h}$ and produces three outputs: $\lambda = f_0(\mathbf{h})$, $f_V(y) = \big[\mathbf{f_V}(\mathbf{h})\big](y)$ and $f_A(y) = \big[\mathbf{f_A}(\mathbf{h})\big](y)$, where f_0 is a scalar output of a fully connected layer with a sigmoid activation, and $\mathbf{f_V}(\mathbf{h})$, $\mathbf{f_A}(\mathbf{h})$ are vector outputs that are used to re-scale the experts predictions according to Eq. (1). The key idea is very simple: We learn to reverse the familiarity effect, by fitting a function from the number of samples that a class has, to a debiasing weight over the confidence for that class.

Formally, $f_V(\mathbf{h})$ uses a fully connected layer to map the embedding vector $\mathbf{h}$ to a $k \times 1$ polynomial coefficients $\mathbf{v}$ that weights a class according to the number of its training samples. Similarly, $f_A(\mathbf{h})$ maps $\mathbf{h}$ to a $k \times 1$ polynomial coefficients $\mathbf{a}$:

$$\mathbf{f_V}(\mathbf{h}, n_y) = \sigma\Big[\sum_{i=0}^{k-1} v_i \big(\frac{n_y}{\max_y (n_y)}\big)^i\Big] \qquad (2)$$

$$\mathbf{f_A}(\mathbf{h}, n_y) = \sigma\Big[\sum_{i=0}^{k-1} a_i \big(\frac{n_y}{\max_y (n_y)}\big)^i\Big] \qquad (3)$$

where n_y is the number of training samples for a class y, k is the polynomial degree and σ denotes a sigmoid that ensures that the resulting scale is positive and non-zero.

2.3. Training the fusion-module

We aim to have the fusion-module learn the relation between the number of training samples and the output confidence (the familiarity bias), so it can adjust for it.

Unfortunately, while the familiarity effect is substantial in the validation data and in the test data, it is not present in the training data. The reason is: Models tend to overfit and become overconfident over rare classes in the training set. This effect is illustrated in Figure 3 (compare Train versus Validation curves). To address this mismatch, we hold-out a subset of the training data and use it to simulate the response of experts to test samples. This set is used for training the fusion-module. We train the architecture with a cross-entropy loss over the outputs of $S(y)$ (Eq. (1)) L_1 normalized by their sum across classes.

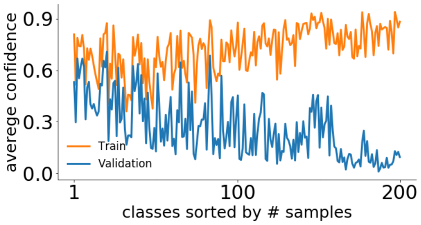

Figure 3: Train-validation mismatch of the familiarity effect: We plot the mean confidence (softmax output) over all samples for each class.

3. Experiments

We evaluate DRAGON in two unbalanced benchmark scenarios. First, *"Smooth-Tail"*, a long-tail setup where the distribution of classes smoothly decays exponentially. Second, *"Two-Level"*, a two-level distribution, as in [12], where a large fraction of classes have many samples, and the rest have few samples. We compare it to SoTA approaches.

The experimental framework is based on [12] and [15, 16], which is the common experimental framework for comparing GZSL methods. Our evaluation uses their features (ResNet-101), and where applicable, also uses their cross-validation splits and evaluation metrics to allow direct comparisons. See implementation details and ablation at `https://arxiv.org/abs/2004.02235`.

Datasets: CUB [14] contains 11,788 visual images of 200 bird species. Each species is described by 312 binary attributes. **AWA** [7] consists of 30,475 images of 50 animal classes and 85 binary attributes. **SUN** [10], contains 14,340 visual scenes, from 717 scene types and 102 attributes.

Visual expert: The visual expert is a logistic regression classifier over pretrained ResNet features to predict $p(y|\mathbf{x})$.

Attribute expert: For the attribute expert, we use LAGO [1], a SoTA ZSL model that models classes as compositions of soft AND-OR expressions across attributes.

Baselines and variants: We compared DRAGON with SoTA approaches and common baselines: *Anchor Loss[11]*, *LDAM Loss[2]* and *CADA-VAE [12]*.

3.1. Smooth-Tail benchmark

To evaluate DRAGON in a long-tail learning scenario we created new variants with long-tail distribution of the three main learning-with-attributes benchmarks - CUB, SUN and AWA. The distribution was created by ranking the classes based on the number of samples and applying an exponential function of the form ab^{-rank}. Here, a and b were selected such that the first class has the maximum number of samples, and the last class has ~2-3 samples.

Evaluation metrics: We evaluated the Smooth-Tail benchmark with: (1) *Per-Class Accuracy* (Acc_{PC}): Balanced accuracy metric that uniformly averages the individual accuracy of each class. (2) *Weighted Accuracy* (Acc_{WGT}): Test accuracy, where the distribution over test classes is long-tailed like the training distribution. This is expected to be the typical case in real-world scenarios.

Results with smooth-tail distribution: Table 1 provides

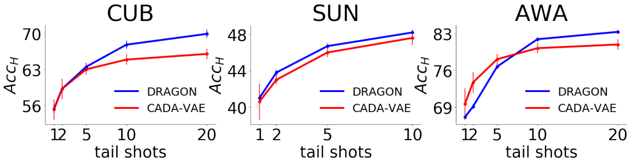

Figure 4: Comparing Acc_H for DRAGON with CADA-VAE, when increasing number of few-shot training samples.

	CUB		SUN		AWA	
	Acc_{PC}	Acc_{WGT}	Acc_{PC}	Acc_{WGT}	Acc_{PC}	Acc_{WGT}
ANCHOR [11]	48.3	64.7	28.2	36.2	59.1	93.2
LDAM [2]	50.1	64.1	29.8	36.4	69.1	93.5
CADA-VAE [12]	48.3	57.4	32.8	35.1	73.5	89.5
DRAGON (OURS)	**57.8**	**67.7**	**34.8**	**40.4**	**74.0**	**94.0**

Table 1: Comparing DRAGON with baselines on the long-tailed benchmark datasets. We report **Per-Class Accuracy** and **Weighted Accuracy**.

the test accuracy for the three long-tail benchmark datasets and compares DRAGON to SoTA baselines. DRAGON achieves higher accuracy compared with all competing methods, both with respect to class-balanced accuracy (Acc_{PC}) and to test-distribution accuracy (Acc_{WGT}).

Improving Acc_{WGT} indicates that DRAGON effectively classifies head classes, which are heavily weighted in Acc_{WGT}. At the same time, improving Acc_{PC} indicates that DRAGON also effectively classifies tail classes, which are up-weighted in Acc_{PC}.

3.2. Two-Level benchmark

The Two-Level benchmark follows the protocol of [12]. For all datasets, many-shot classes have remained the same as in the train-set while few-shot classes have increasing number of shots: 1,2,5,10 and 20 (in SUN up to 10 shots).
Evaluation metrics: We evaluated the Two-Level benchmark with standard metrics from [12]. We report Acc_H, the harmonic mean accuracy over many-shot classes and few-shot classes.

Results with two-level distribution: Figure 4 compares DRAGON with CADA-VAE, a SoTA approach for GFSL-a. DRAGON provides equivalent or better results compared to CADA-VAE although DRAGON is much simpler to train and tune. Importantly, DRAGON excels when the number of shots increases.

4. Extension to vision-only long-tail learning

A simpler variant of the model can be applied to learning with visual samples only. Here, the fusion-module only learns to debias the predictions of a visual model. We name it smDRAGON for *single-modality*-DRAGON and show that it achieves new state-of-the-art results on long-tail CIFAR-10 and CIFAR-100 [6].

To adapt to single-modality, we train smDRAGON to re-scale the predictions of a visual expert using a simplified fusion-module that outputs a single set of coefficients $\{f_V(y)\}_{y \in \mathcal{Y}}$ instead of two sets of coefficients. Subsequently, Eq. 1 reduces to $S(y) = f_V(y)p_V(y)$.

DATASET	UNBALANCED CIFAR-10				UNBALANCED CIFAR-100			
IMBALANCE TYPE	LONG-TAIL		TWO-LEVEL		LONG-TAIL		TWO-LEVEL	
IMBALANCE RATIO	100	10	100	10	100	10	100	10
CB RESAMPLE [3]	29.4	13.2	38.1	15.4	66.5	44.9	66.2	46.9
CB REWEIGHT [3]	27.6	13.4	38.0	16.2	66.0	42.8	78.6	47.5
CB FOCAL [3]	25.4	12.9	39.7	16.5	63.9	42.0	80.2	49.9
CE [2]	29.6	13.6	36.7	17.5	61.6	44.3	61.4	45.3
CE*	29.8	13.1	36.6	17.7	61.7	43.7	61.5	45.7
FOCAL [8]	29.6	13.3	36.0	16.3	61.5	44.2	61.4	46.5
LDAM [2]	26.6	13.0	33.4	15.0	60.4	43.0	60.4	43.7
CE* SMDRAGON	**22.0**	**12.1**	**27.1**	**12.3**	**58.0**	**42.2**	**54.4**	**40.9**

Table 2: Top-1 error rate of ResNet-32 on unbalanced CIFAR-10 and CIFAR-100, comparing smDRAGON and state-of-the-art techniques. Rows with * denote reproduced results using code published by [2].

We again tested two distributions: long-tail and two-level, each with two different imbalance ratios $p = 100$ and $p = 10$, as in [2]. We compared with popular baselines and techniques which have been widely used to address unbalanced datasets: *CE* (standard cross-entropy loss), *CB Resample* [3], *CB Reweight* [3], *Focal Loss [8]* and *LDAM Loss [2]*. Table 2 shows that smDRAGON outperforms all baselines and achieves new SoTA results just by using a ResNet-32, trained using a traditional cross-entropy loss as its backbone.

References

[1] Atzmon, Y., et al.: Probabilistic and-or attribute grouping for zero-shot learning. UAI (2018)

[2] Cao, K., et al.: Learning imbalanced datasets with label-distribution-aware margin loss. NIPS (2019)

[3] Cui, Y., et al.: Class-balanced loss based on effective number of samples. CVPR (2019)

[4] Horn, V., et al.: The devil is in the tails: Fine-grained classification in the wild. arXiv preprint arXiv:1709.01450 (2017)

[5] Kang, B., , et al.: Decoupling representation and classifier for long-tailed recognition. ICLR (2020)

[6] Krizhevsky, A.: Learning multiple layers of features from tiny images (2009)

[7] Lampert, et al.: Learning to detect unseen object classes by between-class attribute transfer. CVPR (2009)

[8] Lin, T.Y., et al.: Focal loss for dense object detection. ICCV (2017)

[9] Liu, Z., et al.: Large-scale long-tailed recognition in an open world. In: CVPR (2019)

[10] Patterson, G., et al.: Sun attribute database: Discovering, annotating, and recognizing scene attributes. CVPR (2012)

[11] Ryou, S., et al.: Anchor loss: Modulating loss scale based on prediction difficulty. ICCV (2019)

[12] Schönfeld, E., et al.: Generalized zero-shot learning via aligned variational autoencoders. CVPR (2019)

[13] Tversky, A., et al.: Availability: A heuristic for judging frequency and probability. Cognitive psychology (1973)

[14] Wah, C., et al.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. rep. (2011)

[15] Xian, Y., Schiele, B., Akata, Z.: Zero-shot learning - the good, the bad and the ugly. In: CVPR (2017)

[16] Xian, Y., et al.: Zero-shot learning - A comprehensive evaluation of the good, the bad and the ugly. TPAMI (2018)