Supplementary:
"PC-GZSL: Prior Correction for Generalized Zero Shot Learning"

S Divakar Bhat   Amit More   Mudit Soni   Bhuvan Aggarwal
Honda R&D Co., Ltd.
Tokyo, Japan

1. Proposed Algorithm

We present the summary of our approach in the form of a pseudo-code in **Algorithm 1**. Proposed approach is quite simple, the overall steps involve computing seen unseen prior $P(\mathcal{Y})$ using total number of samples n_s and n_u for respective case. Similarly, using ground truth data, the number of samples for each class are computed to estimate $P^t(y)$. Finally, model bias or effective prior is computed as an empirical average of $P^m(y|x_i)$ for x_i in validation dataset using original trained model. The last step involves, tuning the parameters α and β which give best harmonic mean accuracy. It is noted that, validation data and test data must have the same prior distribution $P^t(y)$.

2. Additional analysis on CUB and AwA2

We show in Figure 1 the confusion between seen and unseen classes. The top row show results for CUB and bottom row is for AwA2 dataset. One may note that, accuracy numbers in the 1st row (+Ve detections) for each case improves when using PC. Similarly, the bias of unseen classes toward

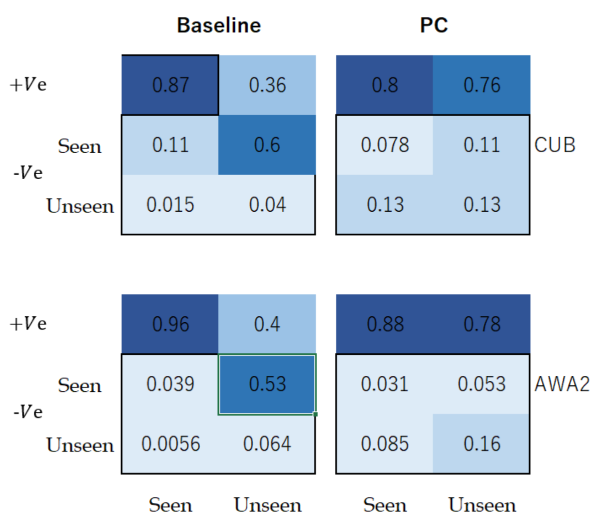

Figure 1. We show the confusion between seen and unseen classes for incorrectly classified examples for the CUB and AwA2 datasets.

Algorithm 1 Algorithm for model prior correction

Input:
Trained model M
Validation data $(x_i, y_i)_{i=1}^{n_s}$ and $(x_i, y_i)_{i=1}^{n_u}$

Output:
Adjusted model M^a

Start:
Compute the adjustment terms

$$P(\mathcal{Y}) = \begin{cases} \frac{n_s}{n_s+n_u} & \text{if } y \in \mathcal{Y}_s \\ \frac{n_u}{n_s+n_u} & \text{otherwise} \end{cases}$$

$$P^t(y) = \frac{1}{n_s+n_u} \sum_i^{n_s+n_u} 1_{hot}(y_i)$$

$$P^m(y) = \frac{1}{n_s+n_u} \sum_i^{n_s+n_u} M(x_i)$$

Tune the adjustment strength

for $(\alpha = 0, \alpha < 2, \alpha += 0.1)$ **do**
  for $(\beta = 0, \beta < 2, \beta += 0.1)$ **do**
    $y_i^{pred} = ArgMax\left(M(x_i).\left(\frac{P^t(y)}{P^m(y)}\right)^{\alpha}.P(\mathcal{Y})^{\beta}\right)$
    h = `Get_Harmonic_Mean_Acc` (y, y^{pred})

    if h is the best
    $(\alpha^*, \beta^*) = (\alpha, \beta)$
  end for
end for

$M^a = \left(\frac{P^t(y)}{P^m(y)}\right)^{\alpha^*} P(\mathcal{Y})^{\beta^*}.M$

Return M^a
End

---

the seen classes (-Ve detections classified into Seen classes) is lower for PC.

3. Ablation on AwA2

Table 1 and Table 2 show results on AwA2 dataset where each component for the adjustment is tested for its effectiveness.

$\log \frac{P^t(y)}{P^m(y)}$	$\log P(\mathcal{Y})$	$\mathcal{U}$	$\mathcal{S}$	$\mathcal{H}$	A
			AwA2		
X	X	34.9	95.3	51.1	63.9
✓	X	71.0	88.1	78.6	82.3
X	✓	46.9	94.0	62.6	69.6
✓	✓	72.7	87.6	**79.5**	**82.7**

Table 1. We show contribution of individual adjustment terms in removing seen unseen class bias for AwA2.

Adjustment	$\mathcal{U}$	$\mathcal{S}$	$\mathcal{H}$	A
		AwA2		
-	34.9	95.3	51.1	63.9
$\log P^t(y)$	59.8	89.5	71.7	80.6
$\log P^m(y)$	44.6	**92.8**	60.2	64.2
$\log \frac{P^t(y)}{P^m(y)}$	**71.0**	88.1	**78.6**	**82.3**

Table 2. The effectiveness of bias removal using $\log P^t(y)$ and $\log P^m(y)$ terms for AwA2.