Hyperspherical Learning in Multi-Label Classification Supplementary Material

Bo Ke[1], Yunquan Zhu[1], Mengtian Li[2], Xiujun Shu[1], Ruizhi Qiao[1], and Bo Ren[1]

[1] Tencent YouTu Lab
{boke, yunquanzhu, ruizhiqiao, timren}@tencent.com, shuxj@mail.ioa.ac.cn
[2] East China Normal University
mtli@stu.ecnu.edu.cn

A Appendix

A.1 Pseudocode

Algorithm 1 provides the pseudo-code of hyperspherical multi-label classification (HML). The loss for hyperspherical multi-label classification includes two parts, one for binary cross-entropy (BCE) with margin based sigmoid and the other for adaptive learning.

A.2 Back-propagation to logits in BCE

In Sec. 3.3, the gradient of the logits is used to analyze the effect of the false negatives. In Sec. 3.4, the total amount of gradients for positives and negatives in different margin m is shown in detail. In this section, we describe how the gradient of logits from BCE loss is calculated. BCE loss composes of positive part $\sum_{i\in\mathcal{Y}}^{N}\log(p_i)$ and negative part $\sum_{i\notin\mathcal{Y}}^{N}\log(1-p_i)$.

$$\mathcal{L}_{\text{BCE}}(p,\mathcal{Y}) = -(\sum_{i\in\mathcal{Y}}^{N}\log(p_i) + \sum_{i\notin\mathcal{Y}}^{N}\log(1-p_i)) \tag{9}$$

where p_i is the probability of the ith class.

$$p = \sigma(s * \cos(\arccos(\frac{Wx}{\|W\|\,\|x\|}) + m)) = \sigma(z) \tag{10}$$

where z is the logit. Here we start from the derivative of p with respect to z,

$$\frac{\partial p}{\partial z} = \sigma(z)(1-\sigma(z)) \tag{11}$$

The derivative of $\mathcal{L}_{\text{BCE}}$ with respect to p is

$$\frac{\partial \mathcal{L}_{\text{BCE}}}{\partial p} = -(\sum_{i\in\mathcal{Y}}^{N}\frac{1}{p_i} - \sum_{i\notin\mathcal{Y}}^{N}\frac{1}{1-p_i}) \tag{12}$$

Algorithm 1 Pseudocode of HML in a PyTorch-like style.

```
# B: Batch size
# L: Length of feature/embedding
# N: Number of labels
# feat:
#    Features extracted from ResNet-50,
#    with shape of (B, L)
# weight:
#    Label embeddings for multiple classes,
#    with shape of (N, L)
# target:
#    Labels with 1 as positives and
#    0 as negatives, shape (B, N)

# learnable parameters
weight = Parameter((N, L))
scale = Parameter(20)
margin = Parameter(0.3)

def train(feat, target):
    # learning in hyperspherical space
    norm_feat = normalize(feat)
    norm_weight = normalize(weight)
    logit = linear(norm_feat, norm_weight)

    # loss for BCE with margin based sigmoid
    m_logit = (logit.acos() + margin).cos() * scale
    loss_bce = BCEWithLogitsLoss(m_logit, target)

    # loss for adaptive learning
    d_logit = logit.detach()
    d_logit = (d_logit.acos() + margin).cos() * scale
    prob = sigmoid(d_logit)
    neg_grad = prob[target == 0].sum()
    pos_grad = (1 - prob[target == 1]).sum()
    loss_adpt = (pos_grad - neg_grad).abs()

    # overall loss
    loss = loss_bce + loss_adpt
    return loss

def test(feat):
    # consistent with training
    norm_feat = normalize(feat)
    norm_weight = normalize(weight)
    logit = linear(norm_feat, norm_weight)
    m_logit = (logit.acos() + margin).cos() * scale
    prob = sigmoid(m_logit)
    return prob
```

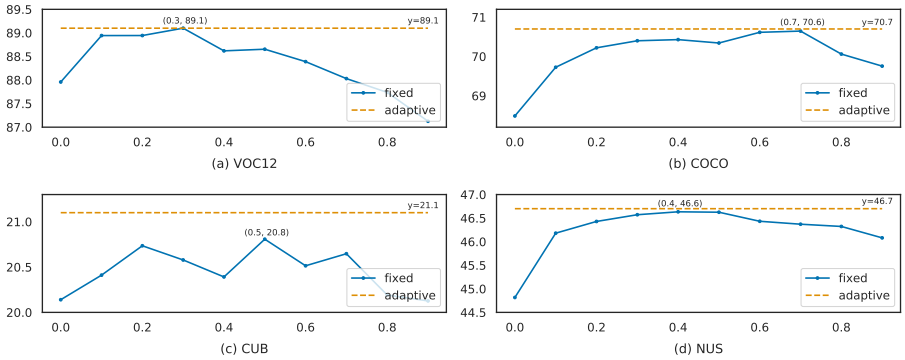


Fig. 7. Experiments on adaptive margin and fixed margin.

Thus the derivative of $\mathcal{L}_{\text{BCE}}$ with respect to z is

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{BCE}}}{\partial z} &= \frac{\partial \mathcal{L}_{\text{BCE}}}{\partial p}\frac{\partial p}{\partial z} \\ &= -\sum_{i\in\mathcal{Y}}^{N}(1-p_i) + \sum_{i\notin\mathcal{Y}}^{N} p_i\end{aligned} \tag{13}$$

For positives, the accumulated gradients in a mini-batch is $\sum_{i\in\mathcal{Y}}^{N}(1-p_i)$. For negatives, the accumulated gradients in a mini-batch is $\sum_{i\notin\mathcal{Y}}^{N} p_i$.

Through gradient analysis, we can find the following two aspects,

- As the margin m increases, the gradient of positives will also become larger, and the gradient of negatives, especially the false negatives, will decrease, thus achieving the effect of suppressing false negatives.
- Margin m can be used to balance the gradient of positives and negatives. Thus, we can optimize the classification by adaptive learning.

A.3 Supplementary Experiments on Adaptive Learning

In Sec. 4.5.2, the experiments illustrate the importance of adaptive learning. To further demonstrate the effectiveness of adaptive learning, we add experiments on fixed margin, as shown in Fig. 7. Specifically, we do a grid search about margin m on four datasets (VOC12, COCO, CUB and NUS). We find that the optimal margin m are different for each dataset. As an example, the optimal margin for the COCO dataset is 0.7. However, this setting would cause a large degradation on VOC12 dataset. In addition, models with adaptive learning outperform those with fixed margin in all datasets with no need to grid searching. From these experiments, it is clear that our method on adaptive learning can reduce the hyper-parameter search while maintaining the performance.

A.4 Noisy Learning

Since the noisy learning methods (Co-teaching[1], SL[2] and JoCoR[3]) only experiment on single-label datasets, such as MNIST. We modify the cross entropy to binary cross entropy to enable these methods working on the multi-label dataset. All experiments are conducted with 50% partial positive labels, so unknown positives are noisy labels. Experiments in Table 6 indicate that these noisy learning methods based on single-label classification do not generalize well enough on multi-label problem, and are less effective than our method.

Table 6. Experiments on COCO with noisy learning methods.

Method	Co-teaching	SL	JoCoR	BCE	HML (Ours)
mAP	70.2	<u>72.0</u>	70.9	69.9	**73.5**

A.5 Label correlation

In Sec. 4.6, we show the label correlation on the COCO dataset. In this section, we additionally present label correlations on the VOC12, CUB and NUS dataset.

VOC12. The label correlation on the VOC12 dataset is shown in Fig. 8. The *person* is the most appearing label on the VOC12, so it is associated with many other labels in learning correlation. In addition, the *chair*, *diningtable* and *sofa* are also related to some extent.

CUB. The CUB dataset contains fine-grained bird attributes, including color, pattern, and shape of a particular part. The label correlation in Fig. 9 illustrates our approach to learn color and texture from the data. The labels with the same color will have a stronger correlation, even they are from different parts of birds. It is because the bird with the same color all over its body would be more common. The same situation occurs in the labels about patterns, such as *has_breast_pattern::spotted* and *has_belly_pattern::spotted*.

NUS. The label correlation on the NUS dataset is shown in Fig. 10. In NUS dataset, there are many pairs of related labels, such as *airport* and *plane*, *boat* and *harbor*, *sky* and *cloud*, *fish* and *coral*. This is because these pairings are more frequently seen in the training data.

From these observations, our approach could implicitly learn label correlation in multi-label classification by learning in hyperspherical space.

A.6 Visualization

In Fig. 11, we visualize the top-5 results of the BCE and HML in the VOC12 datasets. On the one hand, our method suppresses the effect of false negatives during the training and thus is capable of increasing the precision in validation set, as in rows 1-5. On the other hand, our method is able to recall more labels

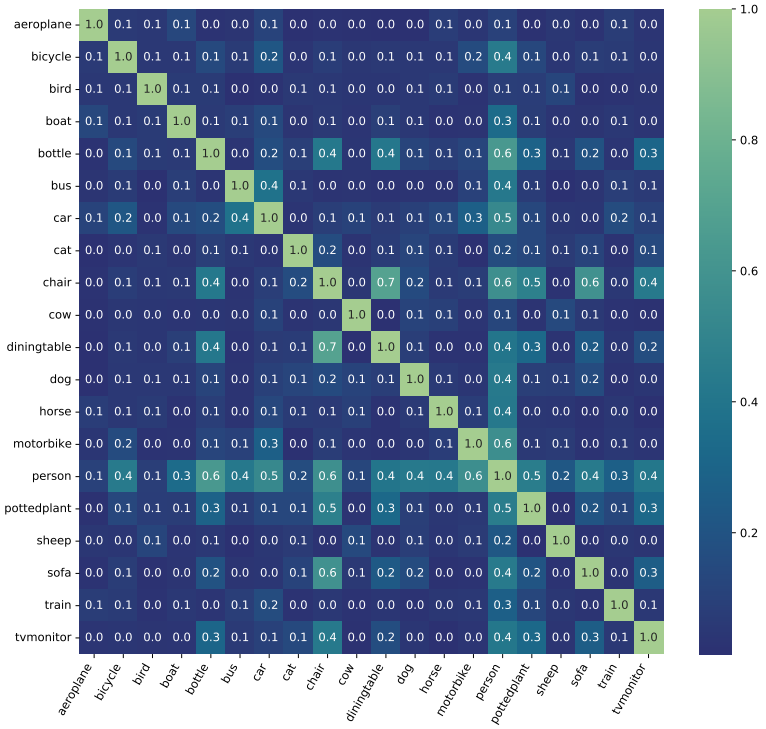


Fig. 8. Label correlation in the VOC12 dataset.

even if the validation set is not completely cleanly annotated, as shown in rows 6-7. These examples illustrate the ability of our method to learn effective features from noisy data.

References

1. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. Advances in neural information processing systems **31** (2018)
2. Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., Bailey, J.: Symmetric cross entropy for robust learning with noisy labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 322–330 (2019)
3. Wei, H., Feng, L., Chen, X., An, B.: Combating noisy labels by agreement: A joint training method with co-regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13726–13735 (2020)

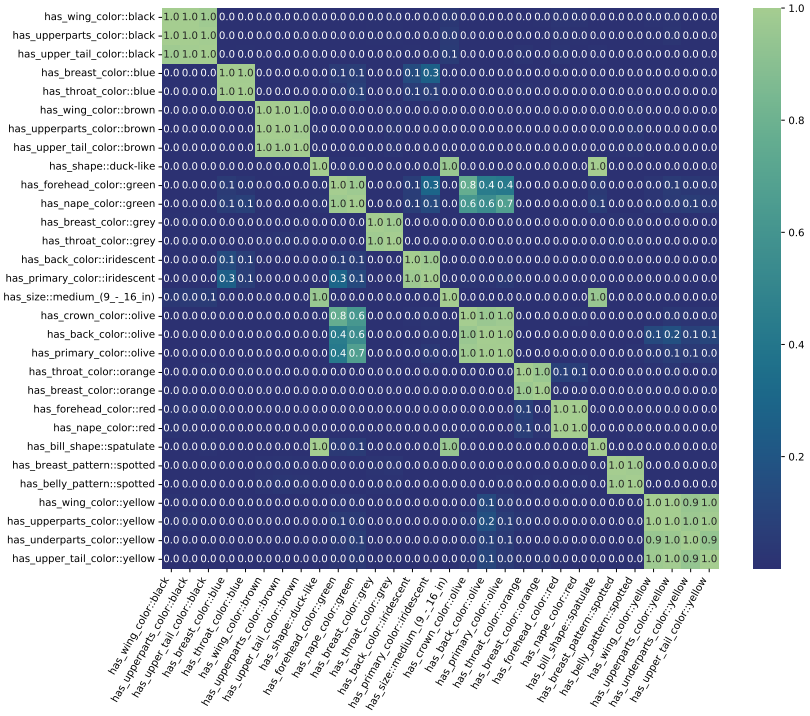


Fig. 9. Label correlation in the CUB dataset.

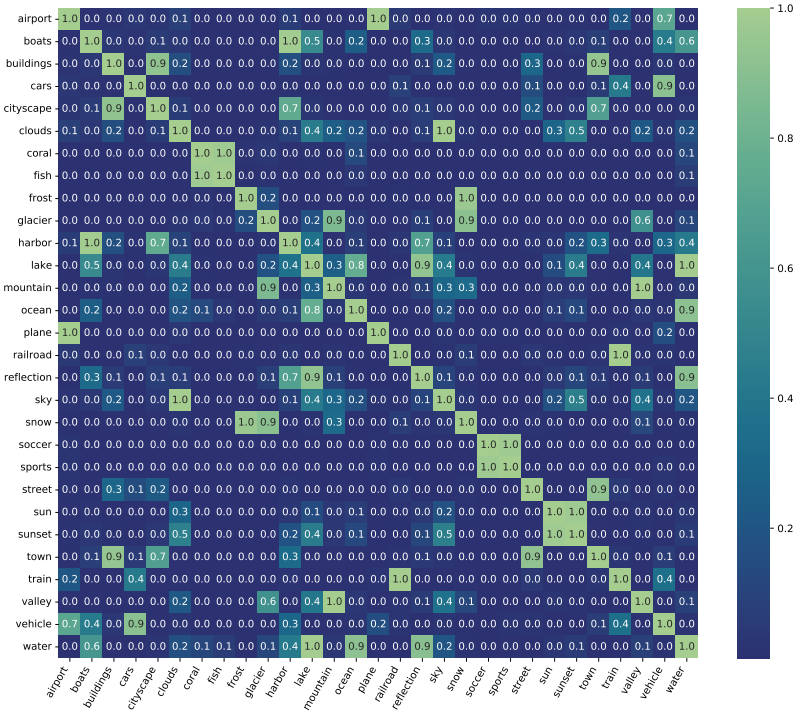


Fig. 10. Label correlation in the NUS dataset.

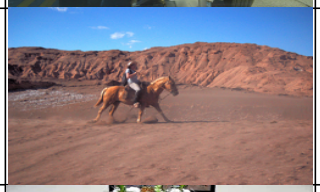
Index	Images	Ground Truth	Top 5 Preds (BCE)	Top 5 Preds (HML)
1		car	boat: 0.5618 car: 0.3349 person: 0.0214 aeroplane: 0.0111 bird: 0.0053	car:0.9197 boat: 0.1021 person: 0.0149 aeroplane: 0.0142 pottedplant: 0.0084
2		sofa	chair: 0.2222 sofa: 0.1960 tvmonitor: 0.1675 pottedplant: 0.0553 diningtable: 0.0370	sofa: 0.6366 chair: 0.1741 tvmonitor: 0.0862 boat: 0.0178 pottedplant: 0.0089
3		bottle person	dog: 0.1418 chair: 0.1399 bottle: 0.1196 diningtable: 0.0986 person: 0.0938	person: 0.4005 bottle: 0.1104 chair: 0.0518 diningtable: 0.0481 dog: 0.0346
4		car dog person	car: 0.9920 person: 0.0289 motorbike: 0.0118 bus: 0.0034 pottedplant: 0.0027	car: 0.9401 person: 0.0831 dog: 0.0181 horse: 0.0103 bus: 0.0074
5		horse person	person: 0.4048 dog: 0.0609 cow: 0.0592 horse: 0.0412 bird: 0.0203	person: 0.5147 horse: 0.4203 cow: 0.0411 dog: 0.0122 bird: 0.0079
6		chair diningtable	person: 0.7090 diningtable: 0.1009 chair: 0.0384 pottedplant: 0.0191 bottle: 0.0099	chair: 0.4707 pottedplant: 0.2367 diningtable: 0.2335 person: 0.0763 sofa: 0.0388
7		cat pottedplant	cat: 0.4264 person: 0.0647 bird: 0.0442 pottedplant: 0.0308 chair: 0.0078	cat: 0.9519 person: 0.0497 pottedplant: 0.0089 bottle: 0.0070 bird: 0.0063

Fig. 11. Visualization on the VOC12 dataset. We list the top 5 predictions for each images. True positives are in Green, while false positives are in Red. True positives that are not in ground truth are marked as Yellow.