Supplementary Material for Multi-Similarity Loss with General Pair Weighting for Deep Metric Learning

Xun Wang, Xintong Han, Weiling Huang*, Dengke Dong, Matthew R. Scott
Malong Technologies, Shenzhen, China
Shenzhen Malong Artificial Intelligence Research Center, Shenzhen, China
{xunwang,xinhan,whuang,dongdk,mscott}@malong.com

1. Introduction

This supplementary material provides more insights into our General Pair Weighting (GPW) framework and Multi-Similarity (MS) loss. First, in Section 2, we revisit three additional existing pair-based loss functions under our GPW framework. Next, in Section 3, we showcase the drawback of direct combination of binomial deviance loss [6] and lifted structure loss [1] (BinLifted). Finally, we analyze the impact of batch-size on MS loss with substantial experiments on CUB200 and SOP datasets.

2. Revisit Pair-based Loss Functions

Here we analyze other pair-based loss functions: N-pairs loss [3], NCA loss [2] and histogram loss [4].

N-pairs Loss. Proposed by Sohn *et al*. [3], N-pairs loss, as a special case of lifted structure loss only considering single positive pair, follows the exactly same analysis process of lifted structure loss in the main paper (Eq. 6-8 in Section 3.2).

NCA Loss. Salakhutdinov *et al*. introduced NCA loss in [2] to learn a nonlinear embedding to optimize the classification performance of the soft-KNN classifier:

$$\begin{aligned}\mathcal{L}_{nca} &:= \sum_{i=1}^{m} log\frac{\sum_{\boldsymbol{y}_k=\boldsymbol{y}_i} e^{S_{ik}}}{\sum_{i=1}^{m} e^{S_{ik}}} \\ &= \sum_{i=1}^{m}\left[log\sum_{\boldsymbol{y}_k=\boldsymbol{y}_i} e^{S_{ik}} - log\sum_{i=1}^{m} e^{S_{ik}}\right].\end{aligned} \tag{1}$$

Then, following GPW framework, the weight of pair $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$, *i.e*., w_{ij}, can be derived from differentiating $\mathcal{L}_{nca}$ with respect to S_{ij}:

$$\begin{aligned}w_{ij}^{+} &= \frac{e^{S_{ij}}}{\sum_{\boldsymbol{y}_k=\boldsymbol{y}_i} e^{S_{ik}}} - \frac{e^{S_{ij}}}{\sum_{i=1}^{m} e^{S_{ik}}} \\ &= \frac{1}{\sum_{\boldsymbol{y}_k=\boldsymbol{y}_i} e^{S_{ik}-S_{ij}}} - \frac{1}{\sum_{i=1}^{m} e^{S_{ik}-S_{ij}}}, \\ w_{ij}^{-} &= \frac{e^{S_{ij}}}{\sum_{i=1}^{m} e^{S_{ik}}} = \frac{1}{\sum_{i=1}^{m} e^{S_{ik}-S_{ij}}}.\end{aligned} \tag{2}$$

We find that in Eq. 2, the weights for positive and negative pairs are determined by their relative similarities compared with the remaining pairs. And as the weight equations show, NCA loss focuses on hard negative pairs and confident positive pairs, or say, pairs that are within neighbor area of the anchor point in the embedding space.

Histogram Loss. Ustinova *et al*. [4] designed a histogram loss based on quadruplets, whose formulation is as below:

$$\begin{aligned}\mathcal{L}_{hist} &:= \sum_{r=1}^{R}(h_r^{-}\sum_{q=1}^{r} h_q^{+}) \\ &= \sum_{r=1}^{R}(\sum_{q=1}^{r} h_q^{+})h_r^{-} \\ &= \sum_{q=1}^{R}(\sum_{r=q}^{R} h_r^{-})h_q^{+}\end{aligned} \tag{3}$$

where R is the dimension of histograms for positive and negative cosine similarities, h_q^{+} is the histogram estimation at node q of positive pairs' cosine similarities, and h_r^{-} is that of negatives at node r.

$$h_r^{+} = \frac{1}{|\mathcal{S}^{+}|}\sum_{\boldsymbol{y}_i=\boldsymbol{y}_j} \delta_{ijr}, \tag{4}$$

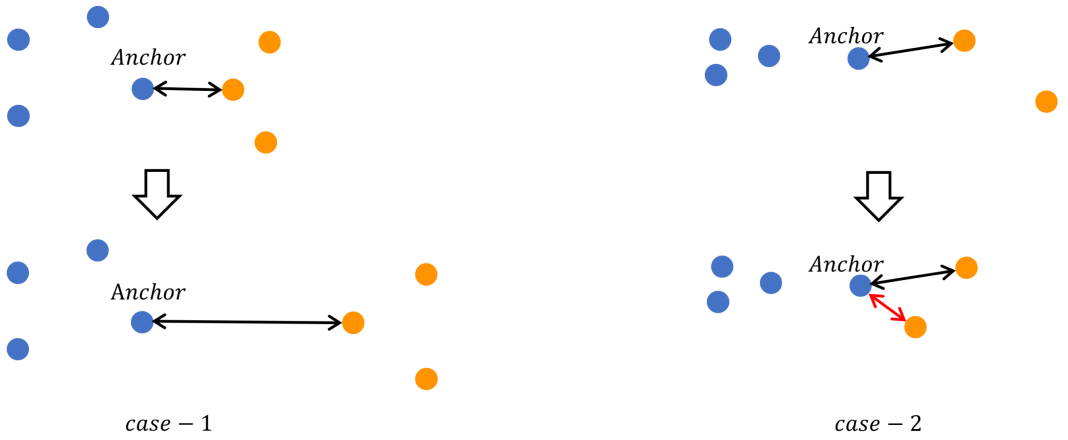

Figure 1. Failure cases of BinLifted weighting scheme: in case-1, the negative pair below is assigned bigger weight than the top one, though its is of much lower cosine similarity; in case-2, one negative is fixed when the other negative sample comes closer to the anchor, resulting the negative pair's *Similarity-N* to be lower.

where δ_{ijr} is defined as:

$$\delta_{ijr} = \begin{cases} (S_{ij} - t_{r-1})/\Delta, & S_{ij} \in [t_{r-1}, t_r], \\ (t_{r+1} - S_{ij})/\Delta, & S_{ij} \in [t_r, t_{r+1}], \\ 0, & otherwise, \end{cases} \tag{5}$$

where $\Delta = 2/(R-1)$, $t_r = r\Delta - 1$. The estimation of h_q^- proceeds analogously.

Before computing the weights assigned for pairs under GPW framework for the complication of histogram loss, we first give the following equations to make the weight calculation more clear (details can be found in [4]).

$$\begin{aligned} \frac{\partial \mathcal{L}_{hist}}{\partial h_q^+} &= \sum_{r=q}^{R} h^- \\ \frac{\partial \mathcal{L}_{hist}}{\partial h_r^-} &= \sum_{q=1}^{r} h^+ \end{aligned} \tag{6}$$

For one positive pair S_{ij}

$$\frac{\partial h_r^+}{\partial S_{ij}} = \begin{cases} \frac{+1}{\Delta|\mathcal{S}^+|}, & S_{ij} \in [t_{r-1}, t_r], \\ \frac{-1}{\Delta|\mathcal{S}^+|}, & S_{ij} \in [t_r, t_{r+1}], \\ 0, & otherwise, \end{cases} \tag{7}$$

where $|\mathcal{S}^+|$ is the number of positive pairs, while we have

$$\frac{\partial h_r^-}{\partial S_{ij}} = 0, \tag{8}$$

since h_r^- is calculated from negative pairs and thus is unrelated to the positive pair's cosine similarity S_{ij}.

Finally, the partial derivative of $\mathcal{L}_{hist}$ w.r.t. one positive pair's similarity $S_{ij} \in [t_p, t_{p+1}]$ is:

$$\begin{aligned} \frac{\partial \mathcal{L}_{hist}}{\partial S_{ij}} &= \sum_{q=1}^{R} \frac{\partial \mathcal{L}_{hist}}{\partial h_q^+} \frac{\partial h_q^+}{\partial S_{ij}} + \sum_{r=1}^{R} \frac{\partial \mathcal{L}_{hist}}{\partial h_r^-} \frac{\partial h_r^-}{\partial S_{ij}} \\ &= \sum_{q=1}^{R} \left(\sum_{r=q}^{R} h_r^- \right) \frac{\partial h_q^+}{\partial S_{ij}} \\ &= \left(\sum_{r=p}^{R} h_r^- \right) \frac{\partial h_p^+}{\partial S_{ij}} + \left(\sum_{r=p+1}^{R} h_r^- \right) \frac{\partial h_{p+1}^+}{\partial S_{ij}} \\ &= \frac{1}{\Delta|\mathcal{S}^+|} \left(\sum_{r=p+1}^{R} h_r^- - \sum_{r=p}^{R} h_r^- \right) \\ &= \frac{-1}{\Delta|\mathcal{S}^+|} h_p^-. \end{aligned} \tag{9}$$

Thus, the weight value assigned to this positive pair is $\frac{1}{\Delta|\mathcal{S}^+|} h_p^-$. Similarly, for one negative pair with cosine similarity $S_{ij} \in [t_p, t_{p+1}]$, its weight under histogram loss is $\frac{1}{\Delta|\mathcal{S}^-|} h_{p+1}^+$.

Though with complicated formulation and rough derivation, the pair weight scheme of histogram loss is extremely concise and clean as shown Eq. 9. h_p^- is approximately the ratio of negative pairs that have lower cosine similarities compared with the current positive pair ($\frac{1}{\Delta|\mathcal{S}^-|}$ can be regarded as a fixed normalizer). Similarly, the weight of one negative pair is the ratio of positive pairs with lower cosine similarity than it. Therefore, the weighting scheme clearly indicts that histogram loss estimates pairs' weights only based on *Similarity-P* (comparing negative (or positive) pairs with positive (or negative respectively) pairs), resulting the poor performance of histogram loss in deep metric learning: it is less effective than binomial deviance loss on CUB200 and SOP as described in [4].

3. BinLifted v.s. MS Weighting

In our ablation study (Section 5.1 in the main paper), we show with experiments that our MS weighting is superior to direct combination of binomial deviance loss and lifted structure loss (BinLifted), though both are based on *Similarities-SN*. Here we explain the benefits of our MS weighting with example of one negative pair $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$. The weight of this negative pair by Binlifted weighting scheme $\widehat{w}_{ij}$ is:

$$\widehat{w}_{ij} = \frac{1}{2}\left(\frac{e^{\beta(S_{ij}-\lambda)}}{1+e^{\beta(S_{ij}-\lambda)}} + \frac{e^{\beta S_{ij}}}{\sum_{\boldsymbol{y}_k \neq \boldsymbol{y}_i} e^{\beta S_{ik}}}\right). \quad (10)$$

From Eq. 10, the weight of BinLifted satisfies:

$$\widehat{w}_{ij} > \frac{1}{2}\max\left(\frac{e^{\beta(S_{ij}-\lambda)}}{1+e^{\beta(S_{ij}-\lambda)}}, \frac{e^{\beta S_{ij}}}{\sum_{\boldsymbol{y}_k \neq \boldsymbol{y}_i} e^{\beta S_{ik}}}\right). \quad (11)$$

Eq. 11 provides a lower bound for weight under BinLifted method. We find that one negative pair with high relative similarity, resulting $\frac{e^{\beta S_{ij}}}{\sum_{\boldsymbol{y}_k \neq \boldsymbol{y}_i} e^{\beta S_{ik}}}$ close to 1, will be assigned with big weight by BinLifted according to Eq. 11 no matter how is its own cosine similarity is. As shown in case-1 of Fig. 1, the weight of the bottom negative pair is close to the top one within BinLifted, though the bottom negative pair is of much lower cosine similarity. Further, when the pair's *Similarity-S* is higher than the threshold λ, lending $\frac{e^{\beta(S_{ij}-\lambda)}}{1+e^{\beta(S_{ij}-\lambda)}}$ to be significantly big, its will be assigned with considerable big weight regardless of its low *Similarity-N*. As illustrated in case-2 of Fig. 1, the two negative pairs at the bottom will be assigned with close weight values according to their high *Similarity-S*, while omitting the huge gap between their relative similarity: *Similarity-N*. However, the negative pair at the top of case-2 will also be assigned with close weight to the bottom one, though its *Similarity* is much higher.

In summary, BinLifted estimates the negative pair's weight mainly depends on the higher one between *Similarity-S* and *Similarity-N*, while neglecting the other one. This drawback lends to its poor performance, even worse than the single binomial deviance loss as shown in the ablation study.

In contrast, its weight assigned by MS weighting is:

$$w_{ij} = \frac{e^{\beta(S_{ij}-\lambda)}}{1+\sum_{\boldsymbol{y}_k \neq \boldsymbol{y}_i} e^{\beta(S_{ik}-\lambda)}}. \quad (12)$$

Therefore, MS weighting is able to update the negative pairs' weights dynamically to address case-1 and case-2 in Fig. 1 through making the best use of the information contained in *Similarities-SN* (Eq. 12), not only focusing on the higher one. Analysis for positive pairs proceeds analogously.

Batch-size	Recall@1 (%)
20	64.75
40	65.07
80	**65.65**
160	65.50
240	64.60

Table 1. Recall@1 performance of MS loss at the batch-size of $\{20, 40, 80, 120, 160, 240\}$ on CUB200.

Batch-size	Recall@1 (%)
20	71.40
40	73.82
80	75.61
160	76.63
320	77.59
640	78.19
1000	**78.35**

Table 2. Recall@1 performance of MS loss at the batch-size of $\{20, 40, 80, 160, 320, 640\}$ on SOP.

4. Effects of Batch Size

To analyze the performance of MS loss at different batch-size, we conduct experiments on the SOP [1] and CUB200 [5] datasets. We set the embedding size as 512 and $K = 5$. We use Adam optimizer with learning rate of 10^{-5} for all experiments. The recall@1 performance of MS loss at the batch-size of $\{20, 40, 80, 160, 240\}$ on CUB200 and the recall@1 results at the batch-size of $\{20, 40, 80, 160, 320, 640, 1000\}$ on SOP are exhibited in Tables 1 and 2.

We observe that batch-size effects the performance of MS loss on CUB200 and SOP quite differently in two folds: (i) CUB200 is less sensitive to the change of batch-size than SOP. (ii) the performance on CUB decreases with larger batch-sizes, while the retrieval result of model trained with MS loss on SOP benefits from large batch-sizes significantly.

This experimental results can be attributed to the fact that CUB200 is a fine-grained dataset with smaller inter-class variations than SOP, resulting higher ratio of hard negative pairs. Moreover, as we observe in experiments at the batch-size of 20 on SOP dataset, there isn't even one informative pairs selected by MS mining at some iteration with a frequency higher than 20%. Therefore, for datasets with low inter-class variations like SOP, we need large batch-sizes (*e.g.*, larger than 320) to obtain considerable informative pairs for training a discriminative model. However, for datasets with small differences between categories like CUB200, the access to enough hard examples is not critical for performance boost.

References

[1] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016. 1, 3

[2] R. Salakhutdinov and G. Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *AISTATS*, 2007. 1

[3] K. Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NIPS*. 2016. 1

[4] E. Ustinova and V. Lempitsky. Learning deep embeddings with histogram loss. In *NIPS*. 2016. 1, 2

[5] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Master's thesis, 2011. 3

[6] D. Yi, Z. Lei, and S. Z. Li. Deep metric learning for practical person re-identification. *arXiv:1407.4979*, 2014. 1