Supplementary Material for "Distilled Person Re-identification: Towards a More Scalable System"

Ancong Wu , Wei-Shi Zheng , Xiaowei Guo , and Jian-Huang Lai
wuancong@gmail.com, wszheng@ieee.org, scorpioguo@tencent.com, stsljh@mail.sysu.edu.cn

Abstract

This supplementary material accompanies our main manuscript "Distilled Person Re-identification: Towards a More Scalable System". More details of teacher similarity matrix construction, training algorithm and implementation are provided.

1. Teacher Similarity Matrix Construction

As illustrated in Section 3.1 in the main manuscript, we need to construct the teacher similarity matrix as the target for the student model to learn.

In practice, when $\mathbf{A}_T$ is not symmetric positive definite (SPD), we can project it onto the cone of all positive semi-definite matrices as in [3]. Let $\mathbf{A}_T = \mathbf{V}_T \mathbf{\Lambda}_T \mathbf{V}_T^\top$ denote the eigendecomposition of $\mathbf{A}_T$. The projection is given by:

$$\mathcal{P}_{\mathcal{S}}(\mathbf{A}_T) = \mathbf{V}_T \mathbf{\Lambda}_T^+ \mathbf{V}_T^\top, \tag{1}$$

where $\mathbf{V}_T$ is the orthonormal matrix of eigenvectors and $\mathbf{\Lambda}_T^+$ is the diagonal matrix of truncated eigenvalues by setting the negative eigenvalues equal to 0. We further make it positive definite by $\mathbf{A}'_T = \mathbf{A}_T + \epsilon \mathbf{I}$, where ϵ is a scalar and $\mathbf{I}$ is an identity matrix.

2. Training Algorithm

The training process of our Multi-teacher Adaptive Similarity Distillation Framework is shown in Algorithm 1. In our implementation, gradients of L_{TA} and L_{VER} with respect to parameters are derived by auto differentiation in PyTorch [1].

Algorithm 1: Training of the Multi-teacher Adaptive Similarity Distillation Framework.

Input : Unlabelled target domain data $\mathcal{D}_U = \{\mathbf{I}_i\}_{i=1}^N$, labelled target domain validation data $\mathcal{D}_L = \{(\mathbf{I}_i, y_i)\}_{i=1}^{N_v}$ of only a few identities, M teacher models $\{H_{Ti}\}_{i=1}^M$ of source domains
Output : Student model H_S, teacher weights $\{\alpha_i\}_{i=1}^M$
Require: Student model learning rate γ, teacher weight learning rate γ_α, simulated feature update step size β

Initialize student model parameter $\mathbf{\Theta}_S$ and initialize teacher weights α_1, α_2,..., α_M as $1/M$
repeat
  Sample batches $\mathcal{B}_U$, $\mathcal{B}_L$ from $\mathcal{D}_U$, $\mathcal{D}_L$, respectively
  Extract features $\mathbf{X}_S^U$, $\mathbf{X}_S^L$ for $\mathcal{B}_U$, $\mathcal{B}_L$ by student model H_S respectively and construct teacher similarity matrices $\{\mathbf{A}_{Ti}\}_{i=1}^M$ by teacher models $\{H_{Ti}\}_{i=1}^M$
  Simulate updating features by Eq. (7) and Eq. (9)
  $\mathbf{X}_S^{U'} = \mathbf{X}_S^U - \beta \frac{\partial L_{TA}(\mathbf{X}_S^U; \{\alpha_i\}_{i=1}^M)}{\partial \mathbf{X}_S^U}$
  $\mathbf{X}_S^{L'} = \mathbf{X}_S^L - \beta \frac{\partial L_{TA}(\mathbf{X}_S^L; \{\alpha_i\}_{i=1}^M)}{\partial \mathbf{X}_S^L}$
  for *each teacher weight* $\alpha_i \in \{\alpha_i\}_{i=1}^M$ **do**
    Update $\alpha_i \leftarrow \alpha_i - \gamma_\alpha \frac{\partial L_{VER}(\mathbf{X}_S^{U'}, \mathbf{X}_S^{L'})}{\partial \alpha_i}$ with Eq. (8) and Eq. (10)
  end
  Update $\mathbf{\Theta}_S \leftarrow \mathbf{\Theta}_S - \gamma \frac{\partial L_{TA}(\mathbf{X}_S^U; \{\alpha_i\}_{i=1}^M)}{\partial \mathbf{\Theta}_S}$ with Eq. (7)
until *student model* H_S *converge*;

3. More Implementation Details

Due to space limitation in Section 5 in the main manuscript, more implementation details are provided here. The teacher similarity matrix can be computed by the cosine similarity of the extracted features as the student similarity matrix as illustrated in Section 3.1 in the main manuscript. In the case of using multiple teachers, the similarities of different cameras and different teachers are not of the same scale because of camera bias and model bias, which hinders effective knowledge aggregation. For the similarities of each camera pair of each teacher, we performed a normalization by dividing the similarities by the mean on training set and in the meantime kept the range of similarities in $[0, 1]$. In testing, the construction of similarity matrix followed the same operation.

The training process consisted of totally 60 epochs. Features of teacher models for target data were extracted only once and stored for reuse to save computation cost. The teacher weights were initialized equally as $1/M$. For the unsupervised setting, the teacher weights were kept $1/M$

[1] https://pytorch.org/

without learning. For the semi-supervised setting, the student model was learned with fixed equal teacher weights α_i in the first 15 epochs, and then dynamic teacher weights were learned by the Adaptive Knowledge Aggregator with labelled data. Batch sizes of unlabelled data batch $\mathcal{B}_U$ and labelled data batch $\mathcal{B}_L$ were 64 and 20, respectively. For the labelled batch for computing validation empirical risk, we sampled two images for each identity to guarantee positive sample pairs. The student model parameter $\mathbf{\Theta}_S$ was trained by ADAM optimizer [2] with initial learning rate $\gamma = 0.001$. The teacher weights α_i were trained by SGD optimizer [1] with momentum 0.9 and initial learning rate $\gamma_\alpha = 0.1$. The learning rates decayed exponentially after 30 epochs. The simulated feature update step β was set as 0.1.

References

[1] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT*. 2

[2] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *Computer Science*, 2014. 2

[3] K. Q. Weinberger, J. Blitzer, and L. K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2005. 1