Deep Transfer Metric Learning

Junlin Hu$^1$, Jiwen Lu$^2$, Yap-Peng Tan$^1$

$^1$School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. $^2$Advanced Digital Sciences Center, Singapore.

How to design a good similarity function plays an important role in many visual recognition tasks. Recent advances have shown that learning a distance metric directly from a set of training examples can usually achieve proposing performance than hand-crafted distance metrics [2, 3]. While many metric learning algorithms have been presented in recent years, there are still two shortcomings: 1) most of them usually seek a single linear distance to transform sample into a linear feature space, so that the nonlinear relationship of samples cannot be well exploited. Even if the kernel trick can be employed to addressed the nonlinearity issue, these methods still suffer from the scalability problem because they cannot obtain the explicit nonlinear mapping functions; 2) most of them assume that the training and test samples are captured in similar scenarios so that their distributions are assumed to be the same. This assumption doesn’t hold in many real visual recognition applications, when samples are captured across datasets.

We propose a deep transfer metric learning (DTML) method for cross-dataset visual recognition. Our method learns a set of hierarchical nonlinear transformations by transferring discriminative knowledge from the labeled source domain to the unlabeled target domain, under which the inter-class variations are maximized and the intra-class variations are minimized, and the distribution divergence between the source domain and the target domain in the top layer of the network is minimized, simultaneously. Figure 1 illustrates the basic idea of the proposed method.

Deep Metric Learning. We construct a deep neural network to compute the representations of each sample $x$. Assume there are $M + 1$ layers of the network and $p^{(m)}$ units in the $m$th layer, where $m = 1, 2, \cdots , M$. The output of $x$ at the $m$th layer is computed as:

$$f^{(m)}(x) = h^{(m)} = \varphi \left( W^{(m)} h^{(m-1)} + b^{(m)} \right) \in \mathbb{R} ^{p^{(m)}},$$

(1)

where $W^{(m)} \in \mathbb{R} ^{p^{(m)} \times p^{(m-1)}}$ and $b^{(m)} \in \mathbb{R} ^{p^{(m)}}$ are the weight matrix and bias of the parameters in this layer; and $\varphi$ is a nonlinear activation function which operates component-wisely, e.g., tanh or sigmoid functions. The nonlinear mapping $f^{(m)} : \mathbb{R} ^d \rightarrow \mathbb{R} ^{p^{(m)}}$ is a function parameterized by $\{ W^{(i)} \}_{i=1} ^{m}$ and $\{ b^{(i)} \}_{i=1} ^{m}$. For the first layer, we assume $h^{(0)} = x$.

For each pair of samples $x_i$ and $x_j$, they can be finally represented as $f^{(m)}(x_i)$ and $f^{(m)}(x_j)$ at the $m$th layer of our designed network, and their distance metric can be measured by computing the squared Euclidean distance between $f^{(m)}(x_i)$ and $f^{(m)}(x_j)$ at the $m$th layer:

$$d^2_{f^{(m)}}(x_i, x_j) = \left\| f^{(m)}(x_i) - f^{(m)}(x_j) \right\|^2_2.$$

(2)

Following the graph embedding framework, we enforce the marginal fisher analysis criterion [4] on the output of all training samples at the top layer and formulate a strongly-supervised deep metric learning method:

$$\min_{f^{(M)}} J = S_c^{(M)} - \alpha S_b^{(M)} + \gamma \sum_{m=1} ^{M} \left( \parallel W^{(m)} \parallel_F + \parallel b^{(m)} \parallel_2^2 \right).$$

(3)

where $\alpha (\alpha > 0)$ is a free parameter which balances the important between intra-class compactness and inter-class separability; $\| Z \|_F$ denotes the Frobenius norm of the matrix $Z$; $\gamma (\gamma > 0)$ is a tunable positive regularization parameter; $S_c^{(m)}$ and $S_b^{(m)}$ define the intra-class compactness and the inter-class separability, which are defined as follows:

$$S_c^{(m)} = \frac{1}{N_{c,1}} \sum_{i=1} ^{N} \sum_{j=1} ^{N} P_{ij} d^2_{f^{(m)}} (x_i, x_j),$$

(4)

$$S_b^{(m)} = \frac{1}{N_{s,1}} \sum_{i=1} ^{N} \sum_{j=1} ^{N} Q_{ij} d^2_{f^{(m)}} (x_i, x_j),$$

(5)

where $P_{ij}$ is set as one if $x_j$ is one of $k_1$-intra-class nearest neighbors of $x_i$, and zero otherwise; and $Q_{ij}$ is set as one if $x_j$ is one of $k_2$-interclass nearest neighbors of $x_i$, and zero otherwise.

Deep Transfer Metric Learning. Given target domain data $X_t$ and source domain data $X_s$, their probability distributions are usually different in the original feature space when they are captured from different datasets. To reduce the distribution difference, we apply the Maximum Mean Discrepancy (MMD) criterion [1] to measure their distribution difference at the $m$th layer, which is defined as follows:

$$D_{ts}^{(m)} (X_t, X_s) = \left\| \frac{1}{N_t} \sum_{i=1} ^{N_t} f^{(m)} (x_i) - \frac{1}{N_s} \sum_{i=1} ^{N_s} f^{(m)} (x_i) \right\|_2 ^2.$$  

(6)

By combining (3) and (6), we formulate DTML as the following optimization problem:

$$\min_{f^{(M)}} J = S_c^{(M)} - \alpha S_b^{(M)} + \gamma D_{ts}^{(M)} (X_t, X_s) + \beta \sum_{m=1} ^{M} \left( \parallel W^{(m)} \parallel_F^2 + \parallel b^{(m)} \parallel_2^2 \right).$$

(7)

where $\beta$ is a regularization parameter. We employ the stochastic gradient descent algorithm to obtain $W^{(m)}$ and $b^{(m)}$.


