Supplementary Material for
Learning Cross-Domain Landmarks for Heterogeneous Domain Adaptation

Yao-Hung Hubert Tsai¹, Yi-Ren Yeh², Yu-Chiang Frank Wang¹
¹Research Center for IT Innovation, Academia Sinica, Taipei, Taiwan
²Department of Mathematics, National Kaohsiung Normal University, Kaohsiung, Taiwan
y.h.huberttsai@gmail.com, yryeh@nknu.edu.tw, ycwang@citi.sinica.edu.tw

We now provide technical details on the derivations of the supervised and full versions of our proposed Cross-Domain Landmark Selection (CDLS) algorithm. For the sake of conciseness, we do not repeat the definitions for each notation in the Supplementary.

I. Optimization of CDLS_{sup}

Recall that in Section 3.2.1 of our manuscript, the objective function of CDLS for heterogeneous domain adaptation (HDA) (i.e., CDLS_{sup}) is:

\[ \min_{A} E_M(A, D_S, D_L) + E_C(A, D_S, D_L) + \lambda \|A\|^2, \quad (i) \]

where \( E_M(A, D_S, D_L) = \) \( \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} A x_s^i - \frac{1}{n_L} \sum_{i=1}^{n_L} \hat{x}_i \right\|^2 \), \( (ii) \) and \( E_C(A, D_S, D_L) = \)

\[ \sum_{c=1}^{C} \frac{1}{n_S^c} \sum_{i=1}^{n_S^c} \sum_{j=1}^{n_L^c} \left\| A x_s^{i,c} - \hat{x}_j \right\|^2 + \]

\[ \frac{1}{n_L^c} \sum_{i=1}^{n_L^c} \sum_{j=1}^{n_L^c} \left\| x_i^{j,c} - \hat{x}_j \right\|^2 \] \( (iii) \).

The minimization problems of (ii) and (iii) with respect to the transformation \( A \) can be rewritten as follows:

\[ A^\top X_S H_{SM,sup} X_S^\top A - 2 A^\top X_S H_{LM,sup} \hat{X}_L^\top + \text{const}, \quad (iv) \]

and

\[ A^\top X_S H_{SC,sup} X_S^\top A - 2 A^\top X_S H_{LC,sup} \hat{X}_L^\top + \text{const}, \quad (v) \]

where

\[ H_{SM,sup}, H_{SC,sup} \in \mathbb{R}^{n_S \times n_S}, \]

\[ H_{LM,sup}, H_{LC,sup} \in \mathbb{R}^{n_S \times n_L}, \]

with entries

\[ (H_{SM,sup})_{ij} = \begin{cases} \frac{1}{n_S^i n_S^j} & \text{if } i, j \in \text{class } c \text{ and } i = j, \\ \frac{1}{n_S^i n_S^j} & \text{if } i, j \in \text{class } c \text{ and } i \neq j, \\ 0 & \text{otherwise}\end{cases} \]

\[ (H_{SC,sup})_{ij} = \begin{cases} \frac{2}{n_S^i n_L^j} & \text{if } i, j \in \text{class } c, \\ 0 & \text{otherwise}\end{cases} \]

By taking the derivatives of (iv) and (v) in (i) with respect to \( A \) and setting it as 0, the closed-form of \( A \) can be derived as:

\[ A = (\lambda I_{d_S} + X_S (H_{SM,sup} + H_{SC,sup}) X_S^\top)^{-1} \left( X_S (H_{LM,sup} + H_{LC,sup}) \hat{X}_L \right). \] 

From the above derivations, the optimal solution \( A \) for our CDLS_{sup} can be obtained.

II. Optimization of CDLS

We now detail the optimization process for the full version of CDLS, which can be applied to solve semi-supervised HDA problems. For simplicity, we have \( \{X_S, X_T, X_U\} \) denote source-domain data, labeled and unlabeled target-domain data, respectively.

II.1. Derivation of Transformation \( A \)

In our work, we apply the technique of alternative optimization for solving CDLS. As noted in our manuscript, with fixed landmark weights \( \alpha \) and \( \beta \), the objective function for solving \( A \) is:
\[ \min_{A} E_M(A, D_S, D_L, X_U, \alpha, \beta) + E_C(A, D_S, D_L, X_U, \alpha, \beta) + \lambda \|A\|^2, \tag{vii} \]

where \(E_M(A, D_S, D_L, X_U, \alpha, \beta) = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \alpha_i A x_i^c - \frac{1}{n_L + n_U} \left( \sum_{i=1}^{n_L} x_i + \sum_{i=1}^{n_U} \beta_i x_i^c \right) \right\|^2. \tag{viii} \]

\[ E_C(A, D_S, D_L, X_U, \alpha, \beta) = \sum_{c=1}^{C} E_{cond}^{c} + \frac{1}{e^c} E_{emb}^{c}, \tag{ix} \]

Recall that, we have
\[ E_{cond}^{c} = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \alpha_i A x_i^c - \frac{1}{n_L + n_U} \left( \sum_{i=1}^{n_L} x_i + \sum_{i=1}^{n_U} \beta_i x_i^c \right) \right\|^2, \]
\[ E_{emb}^{c} = \sum_{i=1}^{n_S} \sum_{j=1}^{n_L} \left\| \alpha_i A x_i^c - \beta_j x_j^c \right\|^2 + \sum_{i=1}^{n_S} \sum_{j=1}^{n_U} \left\| \beta_j x_j^c - \beta_j x_j^c \right\|^2 + \left\| \beta_i x_i^c - \alpha_j A x_j^c \right\|^2, \]

and \(e^c = \delta n_S n_L + \delta n_U n_U + \delta^2 n_U n_S\).

With the constraint of \( \frac{\alpha^T}{n_S} = \frac{\beta^T}{n_U} = \delta\), we rewrite (viii) and (ix) into the following formulations:
\[ A^\top X_S H_{SM} X_S^\top A - 2 A^\top X_S H_{LM} \hat{X}_L^\top - 2 A^\top X_S H_{UM} \hat{X}_U^\top + \text{const}, \tag{x} \]
and
\[ A^\top X_S H_{SC} X_S^\top A - 2 A^\top X_S H_{UC} \hat{X}_U^\top - 2 A^\top X_S H_{UC} \hat{X}_U^\top + \text{const}, \tag{xi} \]

where
\[ H_{SM} = \frac{1}{\delta n_S n_S} \alpha \cdot \alpha^\top, \]
\[ H_{LM} = \frac{1}{\delta n_S (n_L + n_U)} \alpha \cdot \beta^\top, \]
\[ H_{UM} = \frac{1}{\delta n_S (n_L + n_U)} \alpha \cdot \beta^\top, \]
\[ H_{SC} = \begin{bmatrix} \hat{H}_{SC} & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & \hat{H}_{SC} & 0 \\ 0 & 0 & 0 & \hat{H}_{SC} \end{bmatrix}, \tag{with} \]
\[ H_{SC}^c = \frac{1}{\delta^2 n_S n_S} \alpha^c \cdot \alpha^c \cdot \alpha^c + \frac{n_L}{e^c} \cdot \text{diag}(\alpha^c \circ \alpha^c), \]
\[ \hat{H}_{SC} = \begin{bmatrix} H_{LC} & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & H_{LC} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \]
\[ H_{LC} = \left( \frac{1}{\delta n_S (n_L + n_U)} + \frac{1}{e^c} \right) \alpha \cdot \beta^\top, \]

and \(\circ\) denotes element-wise multiplication.

Similar to the optimization of CDLS sup, we take the derivatives of (x) and (xi) in (vii) with respect to \(A\) and set it as 0. Then, we obtain the optimal solution \(A\) as:
\[ A = \left( \lambda L + X_S H_S X_S^\top \right)^{-1} \left( X_S \left( H_L \hat{X}_L^\top + H_U \hat{X}_U^\top \right) \right), \tag{xii} \]

where
\[ H_S = H_{SM} + H_{SC}, \]
\[ H_L = H_{LM} + H_{LC}, \]
\[ H_U = H_{UM} + H_{UC}. \]

## II.2. Derivations of Landmark Weights

Now, we fix \(A\) for optimizing landmark weights \(\alpha\) and \(\beta\). The objective function can be formulated as:
\[ \min_{\alpha, \beta} E_M(A, D_S, D_L, X_U, \alpha, \beta) + E_C(A, D_S, D_L, X_U, \alpha, \beta) \tag{xiii} \]

s.t. \(\{\alpha_i, \beta_i\} \in [0, 1], \frac{\alpha^T}{n_S} = \frac{\beta^T}{n_U} = \delta, \)

where \(E_M\) and \(E_C\) are defined in Section II.1.

To solve the above optimization problem, a Gram matrix \(G\) for describing cross-domain data is introduced:
\[ G \in \mathbb{R}^{(n_S + n_L)^2} \times (n_S + n_L + n_U) \]
\[ = [X_S, X_T, X_U]^\top [X_S, X_T, X_U]. \]

For the ease of the derivations, we define four scaling matrices \(\Theta_{M, C1, C2, C3}\) as follows:
\[ \Theta_M = \theta_M \theta_M^\top \text{ with } \theta_M = \left[ \frac{1}{\delta n_S} \cdot \frac{1}{n_L + n_U} \right], \]
\[ \Theta_{C1} = \sum_{c=1}^{C} \theta_{C1}^c \theta_{C1}^c \top \text{ with} \]
\[ \theta_{C1}^c = [0; \cdots; 0; \frac{1}{\delta n_S^c} 1_{n_S}; 0; \cdots; 0; 0; \cdots; 0; \frac{1}{n_L^c + \delta n_U^c} 1_{n_L}; 0; \cdots; 0; 0; \cdots; 0; \frac{1}{n_L^c + \delta n_U^c} 1_{n_L}; 0; \cdots; 0 ] , \]
\[ \Theta_{C2} = \sum_{c=1}^{C} \frac{1}{e} \theta_{C2}^c \theta_{C2}^c \top \text{ with} \]
\[ \theta_{C2}^c = [0; \cdots; 0; 1_{n_S}; 0; \cdots; 0; 0; \cdots; 0; 1_{n_U}; 0; \cdots; 0 ; 0; \cdots; 0 ], \]
\[ \Theta_{C3} = \sum_{c=1}^{C} \text{diag}(\theta_{C3}^c) \text{ with} \]
\[ \theta_{C3}^c = [0; \cdots; 0; \frac{n_L^c + n_U^c}{e} 1_{n_S}; 0; \cdots; 0; 0; \cdots; 0; \frac{n_L^c + n_U^c}{e} 1_{n_L}; 0; \cdots; 0; 0; \cdots; 0; \frac{n_L^c + n_U^c}{e} 1_{n_U}; 0; \cdots; 0 ] . \]

By integrating the scaling matrix with the Gram matrix \( G \), we have the following Gram matrices \( G_{1-3} \) calculated as:
\[ G_1 = G \odot (\Theta_M + \Theta_{C1} + \Theta_{C3}) , \]
\[ G_2 = G \odot (\Theta_M + \Theta_{C1} + \Theta_{C2}) , \]
\[ G_3 = G \odot (\Theta_M + \Theta_{C1} - \Theta_{C2}) . \]

With (xiv), the original optimization problem of (xiii) becomes:
\[ \min_{\alpha, \beta} \frac{1}{2} \alpha^\top K_{S,S} \alpha + \frac{1}{2} \beta^\top K_{U,U} \beta - \alpha^\top K_{S,U} \beta - k_{S,L}^\top \alpha + k_{U,L}^\top \beta + \text{const.} \]  
\[ \text{s.t.} \quad \{ \alpha_i^c, \beta_i^c \} \in [0,1], \quad \frac{\alpha_i^c}{n_S^c} = \frac{\beta_i^c}{n_U^c} = \delta, \]  
\[ (xv) \]

where
\[ K_{S,S} = G_1 (1: n_S, 1: n_S) , \]
\[ K_{U,U} = G_1 (n_S + n_L + 1: \text{end}, n_S + n_L + 1: \text{end}) , \]
\[ K_{S,U} = G_2 (1: n_S, n_S + n_L + 1: \text{end}) , \]
\[ k_{S,L}(i) = \sum_{j=1}^{n_L} G_2 (i, n_S + j) , \]
\[ k_{U,L}(i) = \sum_{j=1}^{n_L} G_3 (n_S + n_L + i, n_S + j) . \]