Supplement to "Data-Driven Approach to Multiple-Source Domain Adaptation"

This supplementary material provides the proofs and some details which are omitted in the submitted paper. The equation numbers in this material are consistent with those in the paper.

6 DERIVATIONS OF ALGORITHM AND RECONSTRUCTION OF JOINT DISTRIBUTION

We now expand the squares of equations (11) and (12) to express the objective only in terms of kernel Gram matrices:

$$\min_{\mathbf{A},\boldsymbol{\gamma}} \sum_{j=1}^{C}\sum_{j'=1}^{C}\gamma_j\gamma_{j'}\frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{i'=1}^{n_t}k(x_i^t,x_{i'}^t)(\mathbf{RA})_{ij}(\mathbf{RA})_{i'j'} - 2\sum_{j=1}^{C}\gamma_j\frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{i'=1}^{n_t}(\mathbf{RA})_{ij}k(x_i^t,x_{i'}^t) \tag{14}$$

$$+\lambda_f[-\sum_{c=1}^{C}2\sum_{k=1}^{n_d}\xi_{k,c}^{new}\sum_{i=1}^{M}\alpha_{i,c}^k k_\mu(\hat{\mu}_{X|Y=c}^i,\hat{\mu}_{X|Y=c}^{new})) + \sum_{k=1}^{n_d}\sum_{k'=1}^{n_d}\xi_{k,c}^{new}\xi_{k',c}^{new}\sum_{i=1}^{M}\sum_{i'=1}^{M}\alpha_{i,c}^k\alpha_{i',c}^{k'}k_\mu(\hat{\mu}_{X|Y=c}^i,\hat{\mu}_{X|Y=c}^{i'})] \tag{15}$$

where k_μ corresponds to the Gaussian kernel function used to perform Kernel PCA, and $(\mathbf{RA})_{:,i}$ denotes the i-th column of $\mathbf{RA}$. We observe that $k_\mu(\hat{\mu}_{X|Y}^{new},\hat{\mu}_{X|Y}^{new})=1$, so we omit this term as well. Here, n_d represents the number of dimensions we use to reconstruct $\Phi(\hat{\mu}_{X|Y=c}^{new})$ on the low-dimensional manifold. For completeness, we fully expand the regularization term:

$$-2\sum_{c=1}^{C}\sum_{k=1}^{n_d}\sum_{i=1}^{M}\sum_{j=1}^{M}\alpha_{i,c}^k\alpha_{j,c}^k k_\mu(\hat{\mu}_{X|Y=c}^i,\hat{\mu}_{X|Y=c}^{new})k_\mu(\hat{\mu}_{X|Y=c}^j,\hat{\mu}_{X|Y=c}^{new})+$$

$$\sum_{k=1}^{n_d}\sum_{k'=1}^{n_d}\sum_{j=1}^{M}\sum_{j'=1}^{M}\alpha_{j,c}^k\alpha_{j',c}^{k'}k_\mu(\hat{\mu}_{X|Y=c}^{new},\hat{\mu}_{X|Y=c}^j)k_\mu(\hat{\mu}_{X|Y=c}^{new},\hat{\mu}_{X|Y=c}^{j'})\sum_{i=1}^{M}\sum_{i'=1}^{M}\alpha_{i,c}^k\alpha_{i',c}^{k'}k_\mu(\hat{\mu}_{X|Y=c}^i,\hat{\mu}_{X|Y=c}^{i'})$$

For simplicity, we assume binary classification and we differentiate with respect to one of the columns of $\mathbf{A}$ (ex. the second row corresponding to the label 1, denoted by $\mathbf{a}_1$). To maintain clarity, we denote the above terms as $T_{1,..,4}$ and we will differentiate each w.r.t $\mathbf{a}_1$. For the first two terms T_1 and T_2 we have:

$$\frac{\partial T_1}{\partial \mathbf{a}_1} = \frac{2}{n_t^2}[\gamma_2\sum_{j'=1}^{C}\gamma_{j'}\sum_{i=1}^{n_t}\sum_{i'=1}^{n_t}k(x_i^t,x_{i'}^t)\mathbf{R}_{i,:}(\mathbf{R}_{i',:}^T\mathbf{a}_{j'}) + \gamma_2^2\sum_{i=1}^{n_t}\sum_{i'=1}^{n_t}k(x_i^t,x_{i'}^t)\mathbf{R}_{i,:}(\mathbf{R}_{i',:}^T\mathbf{a}_1)] \tag{16}$$

$$\frac{\partial T_2}{\partial \mathbf{a}_1} = 2\gamma_2\frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{i'=1}^{n_t}k(x_i^t,x_{i'}^t)\mathbf{R}_{i,:} \tag{17}$$

$$\tag{18}$$

We now address the second two terms (T_3 and T_4) that correspond to the regularizer:

$$\frac{\partial T_3}{\partial \mathbf{a}_1} = -2\sum_{i=1}^{M}\sum_{i'=1}^{M}[\frac{\partial k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{i}_{X|Y=c})}{\partial \mathbf{a}_1} k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{i'}_{X|Y=c}) + \tag{19}$$

$$\frac{\partial k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{i'}_{X|Y=c})}{\partial \mathbf{a}_1} k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{i}_{X|Y=c})]\sum_{k=1}^{n_d}\alpha^k_{i,c}\alpha^k_{i',c} \tag{20}$$

$$\frac{\partial T_4}{\partial \mathbf{a}_1} = \sum_{k=1}^{n_d}\sum_{k'=1}^{n_d}\sum_{i=1}^{M}\sum_{i'=1}^{M}\sum_{j=1}^{M}\sum_{j'=1}^{M}\alpha^k_{i,c}\alpha^k_{i',c}\alpha^{k'}_{j}\alpha^{k'}_{j',c}[[\frac{\partial k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{i}_{X|Y=c})}{\partial \mathbf{a}_1} k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{j}_{X|Y=c}) + \tag{21}$$

$$\frac{\partial k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{j}_{X|Y=c})}{\partial \mathbf{a}_1} k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{i}_{X|Y=c})] \tag{22}$$

where:

$$\frac{\partial k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{i}_{X|Y=c})}{\partial \mathbf{a}_1} = -\frac{1}{2\sigma^2}(\frac{(\hat{\mu}^{new}_{X|Y=c})^T(\hat{\mu}^{i}_{X|Y=c})}{\partial \mathbf{a}_1} + \frac{(\hat{\mu}^{new}_{X|Y=c})^T(\hat{\mu}^{new}_{X|Y=c})}{\partial \mathbf{a}_1})k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{i}_{X|Y=c}) \tag{23}$$

$$\tag{24}$$

and:

$$\frac{(\hat{\mu}^{new}_{X|Y=c})^T(\hat{\mu}^{i}_{X|Y=c})}{\partial \mathbf{a}_1} = \frac{\partial[\frac{1}{n^i_1 n_t}\mathbf{1}^T\mathbf{K}^{i,c}_1\text{diag}((\mathbf{RA})_{:,1})\mathbf{1}]}{\partial \mathbf{a}_1} \tag{25}$$

$$= \frac{\partial[\frac{1}{n^i_1 n_t}\mathbf{1}^T\mathbf{K}^{i,c}_1(\mathbf{RA})_{:,1}]}{\partial \mathbf{a}_1} = \frac{1}{n^i_1 n_t}(\mathbf{K}^{i,c}_1\mathbf{R})^T\mathbf{1} \tag{26}$$

$$\frac{(\hat{\mu}^{new}_{X|Y})^T(\hat{\mu}^{new}_{X|Y})}{\partial \mathbf{a}_1} = \frac{\partial[\frac{1}{n_t^2}\mathbf{1}^T\text{diag}((\mathbf{RA})_{:,1}\mathbf{K}_t\text{diag}((\mathbf{RA})_{:,1})\mathbf{1}]}{\partial \mathbf{a}_1} \tag{27}$$

$$= \frac{\partial[\frac{1}{n_t^2}((\mathbf{RA})_{:,1})^T\mathbf{K}_t(\mathbf{RA})_{:,1}]}{\partial \mathbf{a}_1} = \frac{1}{n_t^2}\frac{\partial[\mathbf{a}_1^T\mathbf{R}^T\mathbf{K}_t\mathbf{R}\mathbf{a}_1]}{\partial \mathbf{a}_1} = \frac{1}{n_t^2}2\mathbf{R}^T K_t\mathbf{R}\mathbf{a}_1 \tag{28}$$

Here, $\mathbf{K}_t$ is the Gram matrix for the kernel mean embedding kernel of the target data, and $\mathbf{K}^{i,c}_1$ is the cross-kernel matrix between the target-domain data, and the data of source i with label 1. n_t is the number of samples in the target domain, and n^i_1 is the number of samples in the i-th source domain that have a label 1. Furthermore, for the mean embeddings of conditional distributions conditioned on discrete labels, we have:

$$\hat{\mu}^{new}_{X|Y=1} = [\frac{1}{n_t}(\psi(\mathbf{x}^T)\text{diag}((\mathbf{RA})_{:,1}))\mathbf{1}] \tag{29}$$

$$\hat{\mu}^{i}_{X|Y=1} = [\frac{1}{n^i_1}\psi(\mathbf{x}^T)\mathbf{1}] \tag{30}$$

6.1 Vectorization of Objective

We rewrite the objective function in terms of matrix operations:

$$\min_{\boldsymbol{\gamma},\mathbf{A}} \frac{1}{n_t^2}\boldsymbol{\gamma}^T F\boldsymbol{\gamma} - \frac{2}{n_t}\boldsymbol{\gamma}^T(\mathbf{K1})^T(\mathbf{RA}) + \lambda_f R(\mathbf{A}) \tag{31}$$

where $\mathbf{F}_{j,j'} = (\mathbf{RA})^T_{:,j}[(\mathbf{K1})\odot(\mathbf{RA})_{:,j'}]$. $R(\mathbf{A})$ represents a regularization term, now given by a summation over labels:

$$\sum_{c=1}^{C}||\Phi(\hat{\mu}^{new}_{X|Y=c}) - P_n\Phi(\hat{\mu}^{new}_{X|Y=c})||^2 = \sum_{c=1}^{C}[k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{new}_{X|Y=c}) - \tag{32}$$

$$2\sum_{k=1}^{n_d}\boldsymbol{\alpha}^{kT}_c\mathbf{D}\boldsymbol{\alpha}^k_c + \sum_{k=1}^{n_d}\sum_{k'=1}^{n_d}(\boldsymbol{\alpha}^{kT}_c\mathbf{D}\boldsymbol{\alpha}^{k'}_c)(\boldsymbol{\alpha}^{kT}_c\tilde{\mathbf{K}}\boldsymbol{\alpha}^{k'}_c)] \tag{33}$$

$k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{new}_{X|Y=c}) = 1$ so it is a constant we can ignore for the optimization. Here, $\mathbf{D}_{ij} = k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{i}_{X|Y=c}) k_\mu(\hat{\mu}^{new}_{X|Y=c}, \hat{\mu}^{j}_{X|Y=c})$, and $\tilde{\mathbf{K}}$ is the Gram matrix over the embeddings of the source domains. Now we write down the derivatives in vectorized form:

$$\frac{\partial T_1}{\partial \mathbf{a}_1} = \frac{2\boldsymbol{\gamma_1}}{n_t}((\mathbf{R}\mathbf{a}_1) \odot (\mathbf{K1})^T\mathbf{R} + \frac{2\boldsymbol{\gamma_1}}{n_t^2}\sum_{j'=1}^{K}\gamma_{j'}((\mathbf{R}\mathbf{a}_{j'}) \odot (\mathbf{K1})^T\mathbf{R} \tag{34}$$

$$\frac{\partial T_2}{\partial \mathbf{a}_1} = \frac{2\boldsymbol{\gamma_1}}{n_t^2}(\mathbf{K1})^T\mathbf{R} \tag{35}$$

$$\frac{\partial T_3}{\partial \mathbf{a}_1} = \sum_{k=1}^{n_d}\alpha_1^{kT}\mathbf{Q}\alpha_1^k \tag{36}$$

$$\frac{\partial T_4}{\partial \mathbf{a}_1} = \sum_{k=1}^{n_d}\alpha_1^{kT}\mathbf{Q}\alpha_1^k\alpha_1^{kT}\tilde{\mathbf{K}}\alpha_1^k \tag{37}$$

where:

$$\mathbf{Q}_{ii'} = [\frac{\partial k_\mu(\hat{\mu}^{new}_{X|Y=1}, \hat{\mu}^{i}_{X|Y=1})}{\partial \mathbf{a}_1}k_\mu(\hat{\mu}^{new}_{X|Y=1}, \hat{\mu}^{i'}_{X|Y=1}) + \frac{\partial k_\mu(\hat{\mu}^{new}_{X|Y=1}, \hat{\mu}^{i'}_{X|Y=1})}{\partial \mathbf{a}_1}k_\mu(\hat{\mu}^{new}_{X|Y=1}, \hat{\mu}^{i}_{X|Y=1})] \tag{38}$$

Having the first derivatives of the objective with respect to colums of $\mathbf{A}$ is sufficient for implementing a barrier method using most packages (we used fmincon() in MATLAB), which then use the first derivatives to approximate the second derivatives during optimization.

7 PROOFS OF THEOREMS AND LEMMATA

7.1 Proof of Lemma 2

We restate the lemma here: *Let points $\phi_1, ..., \phi_M$, be p-dimensional vectors (where p could be infinite). Let $\lambda_1, ..., \lambda_q$ be the set of all non-zero eigenvalues after performing PCA on these vectors. If $P_\lambda(\phi_i)$ is the projection of ϕ_i on the principal eigenvectors corresponding to $\lambda_1, ..., \lambda_q$, then $\phi_i \neq \phi_j \iff P_\lambda(\phi_i) \neq P_\lambda(\phi_j)$*
Proof: Let $\phi_i \neq \phi_j$. Then expressing ϕ_i and ϕ_j in terms of a Fourier expansion on eigenvectors corresponding to the nonzero eigevalues $(\mathbf{v}_1, ..., \mathbf{v}_q)$, we obtain $P_\lambda(\phi_i) = \sum_{k=1}^{q}(\phi_i^T\mathbf{v}_k)\mathbf{v}_k$, $P_\lambda(\phi_j) = \sum_{k=1}^{q}(\phi_j^T\mathbf{v}_k)\mathbf{v}_k$, and there are no residual terms because the rest of the eigenvalues are zero. Then it immediately follows that $P_\lambda(\phi_i) \neq P_\lambda(\phi_j)$.

7.2 Proof of Theorem 2

We restate the theorem here:
Theorem 2 *Let $\mathbf{A}_1$ and $\mathbf{A}_2$ hold, and $\hat{\eta_c}$ be the weights such that $P^{new}_{X|Y=c} = P^{\hat{\boldsymbol{\eta}}_c}_{X|Y=c}$ is the reconstructed distribution, namely $P^{new}_{X|Y=c} = \mathbf{B}_{:,c}P^{\mathcal{T}}_X$. If $\exists\ \hat{\boldsymbol{\eta}}_c$ s.t $P^{\mathcal{T}}_X = \sum_{c=1}^{C}P^{new}_Y(Y=c)P^{\hat{\boldsymbol{\eta}}_c}_{X|Y=c} = \sum_{c=1}^{C}\boldsymbol{\gamma}_c P^{\hat{\boldsymbol{\eta}}_c}_{X|Y=c}$, then we have $\forall\ c$, $P^{\mathcal{T}}_Y(Y=c) = \boldsymbol{\gamma}_c$ and $P^{\hat{\boldsymbol{\eta}}_c}_{X|Y=c} = P^{\mathcal{T}}_{X|Y=c}$.*
Proof: Making use of $\mathbf{A}_1$ to express $P^{\mathcal{T}}_X$ as $P^{\mathcal{T}}_X = \sum_{c=1}^{C}P^{\mathcal{T}}_Y(Y=c)P^{\boldsymbol{\eta}^*_c}_{X|Y=c}$. After solving (6), we can achieve:
$P^{\mathcal{T}}_X = P^{new}_X \implies \sum_{c=1}^{C}P^{\mathcal{T}}_Y(Y=c)P^{\boldsymbol{\eta}^*_c}_{X|Y=c} = \sum_{c=1}^{C}\boldsymbol{\gamma}_c P^{\hat{\boldsymbol{\eta}}_c}_{X|Y=c} \implies$
$\sum_{c=1}[\boldsymbol{\gamma}_c P^{\hat{\boldsymbol{\eta}}_c}_{X|Y=c} - P^{\mathcal{T}}_Y(Y=c)P^{\boldsymbol{\eta}^*_c}_{X|Y=c}] = 0$. By $\mathbf{A}_2$ we can conclude that $\forall\ c$, $\boldsymbol{\gamma}_c P^{\hat{\boldsymbol{\eta}}_c}_{X|Y=c} - P^{\mathcal{T}}_Y(Y=c)P^{\boldsymbol{\eta}^*_c}_{X|Y=c} = 0$.
Taking the integral of the last expression yields $P^{\mathcal{T}}_Y(Y=c) = \boldsymbol{\gamma}_c$, and therefore $P^{\hat{\boldsymbol{\eta}}_c}_{X|Y=c} = P^{\boldsymbol{\eta}^*_c}_{X|Y=c} = P^{\mathcal{T}}_{X|Y=c}$.

8 HYPERPARAMETER TUNING

Here we describe the hyperparameter settings in the baselines and the proposed method in the experiments. For the "poolSVM" method, we tuned the slackness parameter and the kernel width using 5-fold cross-validation on the pooled training data. For the "marg-kernel" method, we used the optimal slackness and k_X kernel width according to cross-validation via "poolSVM". For the k_P kernel, we varied the kernel width on a grid of values proportional to the median pairwise distance of the training data pooled together. For each dataset, we present

the results with a k_P kernel width with the best average performance. For the methods "dist-weight" and "dist-comb", for all kernel mean embeddings we also tried several different values proportional to the median pairwise distance of the training data, and we report the results corresponding to the kernel width with best average performance for each respective baseline.

For our method, we set the kernel mean embedding kernel widths corresponding to k and k_μ, in the same manner as "dist-weight" and "dist-comb",with the only difference that we used only 5 percent of the experiments in each dataset in order to pick kernel widths with the best average performance, and then report the accuracies for all experiments using the selected kernel widths. Regarding the other hyperparameters of our method, in all experiments we fixed σ_B to the median pairwise distance of the data in the target domain. We kept $\lambda_f = 1$ and $\lambda_R = 0.1$.