Optimal Bayesian Transfer Learning

Alireza Karbalayghareh, *Student Member, IEEE,* Xiaoning Qian, *Senior Member, IEEE,*
and Edward R. Dougherty, *Fellow, IEEE*

Abstract—Transfer learning has recently attracted significant research attention, as it simultaneously learns from different source domains, which have plenty of labeled data, and transfers the relevant knowledge to the target domain with limited labeled data to improve the prediction performance. We propose a Bayesian transfer learning framework where the source and target domains are related through the joint prior density of the model parameters. The modeling of joint prior densities enables better understanding of the "transferability" between domains. We define a joint Wishart density for the precision matrices of the Gaussian feature-label distributions in the source and target domains to act like a bridge that transfers the useful information of the source domain to help classification in the target domain by improving the target posteriors. Using several theorems in multivariate statistics, the posteriors and posterior predictive densities are derived in closed forms with hypergeometric functions of matrix argument, leading to our novel closed-form and fast Optimal Bayesian Transfer Learning (OBTL) classifier. Experimental results on both synthetic and real-world benchmark data confirm the superb performance of the OBTL compared to the other state-of-the-art transfer learning and domain adaptation methods.

Index Terms—Transfer learning, domain adaptation, optimal Bayesian transfer learning, optimal Bayesian classifier

I. INTRODUCTION

A basic assumption of traditional machine learning is that data in the training and test sets are independently sampled in one domain with the identical underlying distribution. However, with the growing amount of heterogeneity in modern data, the assumption of having only one domain may not be reasonable. Transfer learning (TL) is a learning strategy that enables us to learn from a source domain with plenty of labeled data as well as a target domain with no or very few labeled data in order to design a better classifier in the target domain than the ones trained by target-only data for its generalization performance. This can reduce the effort of collecting labeled data for the target domain, which might be very costly, if not impossible. Due to its importance, there has been ongoing research on the topic of transfer learning and many surveys in the recent years covering transfer learning and domain adaptation methods from different perspectives [1], [2], [3], [4], [5].

If we train a model in one domain and directly apply it in another, the trained model may not generalize well, but if the domains are related, appropriate transfer learning and domain adaptation methods can borrow information from all the data across the domains to develop better generalizable models in the target domain. Transfer learning in medical genomics is desirable, since the number of labeled data samples is

often very limited due to the difficulty of having disease samples and the prohibitive costs of human clinical trials. However, it is relatively easier to obtain gene-expression data for cell lines or other model species like mice or dogs. If these different life systems share the same underlying disease cellular mechanisms, we may utilize data in cell lines or model species as our source domain to develop transfer learning methods for more accurate human disease prognosis in the target domain [6], [7].

A. Related Works

Domain adaptation (DA) is a specific case of transfer learning where the source and target domains have the same classes or categories [2], [3], [5]. DA methods either adapt the model learned in the source domain to be applied in the target domain or adapt the source data so that the distribution can be close to the one of the target data. Depending on the availability of labeled target data, the DA methods are categorized as unsupervised and semi-supervised algorithms. Unsupervised DA problems applies to the cases where there are no labeled target data and the algorithm uses only unlabeled data in the target domain along with source labeled data [8]. Semi-supervised DA methods use both the unlabeled and a few labeled target data to learn a classifier in the target domain with the help of source labeled data [9], [10], [11], [12].

Depending on whether the source and target domains have the same feature space with the same feature dimension, there are homogeneous and heterogeneous DA methods. The first direction in homogeneous DA is instance re-weighting, for which the most popular measure to re-weight the data is Maximum Mean Discrepancy (MMD) [13] between the two domains. Transfer Adaptive Boosting (TrAdaBoost) [14] is another method that adaptively sets the weights for the source and target samples during each iteration based on the relevance of source and target data to help train the target classifier. Another direction is model or parameter adaptation. There are several efforts to adapt the SVM classifier designed in the source domain for the target domain, for example, based on residual error [15], [16]. Feature augmentation methods, such as Geodesic Flow Sampling (GFS) and Geodesic Flow Kernel (GFK) [8], derive intermediate subspaces using Geodesic flows, which interpolate between the source and target domains. Finding an invariant latent domain in which the distance between the empirical distributions of the source and target data is minimized is another direction to tackle the problem of domain adaptation, such as Invariant Latent Space (ILS) in [17]. Authors in [17] proposed to learn an invariant latent Hilbert space to address both the unsupervised and semi-supervised DA problems, where a notion of domain variance is

The authors are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA.

simultaneously minimized while maximizing a measure of discriminatory power using Riemannian optimization techniques. Max-Margin Domain Transform (MMDT) [10] is a semi-supervised feature transformation DA method which uses a cost function based on the misclassification loss and jointly optimizes both the transformation and classifier parameters. Another domain-invariant representation method [18] matches the distributions in the source and target domains via a regularized optimal transportation model. Heterogeneous Feature Augmentation (HFA) [9] is a heterogeneous DA method which typically embeds the source and target data into a common latent space prior to data augmentation.

Domain adaption has been recently studied in deep learning frameworks like deep adaptation network (DAN) [19], residual transfer networks (RTN) [20], and models based on generative adversarial networks (GAN) such as domain adversarial neural network (DaNN) [21] and coupled GAN (CoGAN) [22]. Although deep DA methods have shown promising results, they require a fairly large amount of labeled data.

B. Main Contributions

This paper treats homogeneous transfer learning and domain adaptation from Bayesian perspectives, a key aim being better theoretical understanding when data in the source domain are "transferrable" to help learning in the target domain. When learning complex systems with limited data, Bayesian learning can integrate prior knowledge to compensate the generalization performance loss due to the lack of data for the required sample complexity based on the predictive models of choice. Rooted in Optimal Bayesian Classifiers (OBC) [23], [24], which minimize the Bayesian error estimates of classifiers over uncertainty classes of feature-label distributions, we propose a Bayesian transfer learning framework and the corresponding Optimal Bayesian Transfer Learning (OBTL) classifier to formulate the OBC in the target domain by taking advantages of both the available data and the joint prior knowledge in source and target domains. In this Bayesian learning framework, transfer learning from the source to target domain is through a joint prior probability density function for the model parameters of the feature-label distributions of the two domains. By explicitly modeling the dependency of the model parameters of the feature-label distribution, the posterior of the target model parameters can be updated via the joint prior probability distribution function in conjunction with the source and target data. Based on that, we derive the *effective* class-conditional densities of the target domain, by which the OBTL classifier is constructed.

Our problem definition is the same as the aforementioned domain adaptation methods, where there are plenty of labeled source data and few labeled target data. The source and target data follow different multivariate Gaussian distributions with arbitrary mean vectors and precision (inverse of covariance) matrices. For the OBTL, we define a joint Normal-Wishart prior distribution, where the two precision matrices in the two domains are jointly connected. This joint prior distribution for the two precision matrices of the two domains acts like a bridge through which the useful knowledge of the source

domain can be transferred to the target domain, making the posterior of the target parameters tighter with less uncertainty.

With such a Bayesian transfer learning framework and several theorems from multivariate statistics, we define an appropriate joint prior for the precision matrices using hypergeometric functions of matrix argument, whose marginal distributions are Wishart as well. The corresponding closed-form posterior distributions for the target model parameters are derived by integrating out all the source model parameters. Having closed-form posteriors facilitates closed-form effective class-conditional densities. Hence, the OBTL classifier can be derived based on the corresponding hypergeometric functions and does not need iterative and costly techniques like MCMC sampling. Although the OBTL classifier has a closed form, computing these hypergeometric functions involves the computation of series of zonal polynomials, which is time-consuming and not scalable to high dimension. To resolve this issue, we use the Laplace approximations of these functions, which preserves the good prediction performance of the OBTL while making it efficient and scalable. The performance of the OBTL is tested on both synthetic data and real-world benchmark image datasets to show its superior performance over state-of-the-art domain adaption methods.

The paper is organized as follows. Section II introduces the Bayesian transfer learning framework. Section III derives the closed-form posteriors of target parameters, via which Section IV obtains the effective class-conditional densities in the target domain. Section V derives the OBTL classifier, and Section VI presents the OBC in the target domain and shows that the OBTL classifier converts to the target-only OBC when there is no interaction between the domains. Section VII introduces the Laplace approximation of the hypergeometric functions of matrix argument. Section VIII presents experimental results using both synthetic and real-world benchmark data. Finally, Section IX concludes the paper.

II. BAYESIAN TRANSFER LEARNING FRAMEWORK

We consider a supervised transfer learning problem in which there are L common classes (labels) in each domain. Let $\mathcal{D}_s$ and $\mathcal{D}_t$ denote the labeled datasets of the source and target domains with the sizes of N_s and N_t, respectively, where $N_t \ll N_s$. Let $\mathcal{D}_s^l = \left\{ \mathbf{x}_{s,1}^l, \mathbf{x}_{s,2}^l, \cdots, \mathbf{x}_{s,n_s^l}^l \right\}, l \in \{1, \cdots, L\}$, denote the n_s^l data of the source domain for the label l. Similarly, let $\mathcal{D}_t^l = \left\{ \mathbf{x}_{t,1}^l, \mathbf{x}_{t,2}^l, \cdots, \mathbf{x}_{t,n_t^l}^l \right\}, l \in \{1, \cdots, L\}$, denote the n_t^l data of the target domain for the label l. Obviously, we have $\mathcal{D}_s = \bigcup_{l=1}^{L} \mathcal{D}_s^l$, $\mathcal{D}_t = \bigcup_{l=1}^{L} \mathcal{D}_t^l$, $N_s = \sum_{l=1}^{L} n_s^l$, and $N_t = \sum_{l=1}^{L} n_t^l$. Since we consider the homogeneous transfer learning scenario, where the feature spaces are the same in both the source and target domains, $\mathbf{x}_s^l$ and $\mathbf{x}_t^l$ are $d \times 1$ vectors for d features of the source and target domains, respectively. We use the Gaussian model for the feature-label distribution in each domain:

$$\mathbf{x}_z^l \sim \mathcal{N} \left(\mu_z^l, \left(\mathbf{\Lambda}_z^l \right)^{-1} \right), \quad l \in \{1, \cdots, L\}, \quad (1)$$

where $z \in \{s, t\}$ denotes the source, s, or target, t, domains; μ_s^l and μ_t^l are the $d \times 1$ mean vectors in the source and target

domains for label l, respectively, and $\mathbf{\Lambda}_s^l$ and $\mathbf{\Lambda}_t^l$ are the $d \times d$ precision matrices in the source and target domains for label l, respectively. In the Bayesian framework, a Normal-Wishart distribution is employed as a prior for mean and precision matrices of the Gaussian models. Here we require a joint prior distribution of the parameters of the source and target domains to account for the dependency or "relatedness" between the domains. We define the following joint prior distribution for μ_s^l, μ_t^l, $\mathbf{\Lambda}_s^l$, and $\mathbf{\Lambda}_s^l$:

$$p\left(\mu_s^l, \mu_t^l, \mathbf{\Lambda}_s^l, \mathbf{\Lambda}_t^l\right) = p\left(\mu_s^l | \mathbf{\Lambda}_s^l\right) p\left(\mu_t^l | \mathbf{\Lambda}_t^l\right) p\left(\mathbf{\Lambda}_s^l, \mathbf{\Lambda}_t^l\right), \quad (2)$$

for $l \in \{1, \cdots, L\}$, where we assume μ_s^l and μ_t^l are conditionally independent given $\mathbf{\Lambda}_s^l$ and $\mathbf{\Lambda}_t^l$ for any class l. For making the priors conjugate, both $p\left(\mu_s^l | \mathbf{\Lambda}_s^l\right)$ and $p\left(\mu_t^l | \mathbf{\Lambda}_t^l\right)$ are considered as Gaussian:

$$\mu_z^l | \mathbf{\Lambda}_z^l \sim \mathcal{N}\left(\mathbf{m}_z^l, \left(\kappa_z^l \mathbf{\Lambda}_z^l\right)^{-1}\right), \quad l \in \{1, \cdots, L\}, \quad (3)$$

for $z \in \{s, t\}$, where $\mathbf{m}_z^l$ is the $d \times 1$ mean vector of μ_z^l, and κ_z^l is a positive scalar hyperparameter. We need to define a joint distribution for $\mathbf{\Lambda}_s^l$ and $\mathbf{\Lambda}_t^l$. In the case of a prior for either $\mathbf{\Lambda}_s^l$ or $\mathbf{\Lambda}_t^l$, we use a Wishart distribution as the conjugate prior. Here we desire a joint distribution for $\mathbf{\Lambda}_s^l$ and $\mathbf{\Lambda}_t^l$, whose marginal distributions for both $\mathbf{\Lambda}_s^l$ and $\mathbf{\Lambda}_t^l$ are Wishart.

Letting $\mathbf{x}^l = \left[\mathbf{x}_t^{l'}, \mathbf{x}_s^{l'}\right]'$ be the $2d \times 1$ augmented feature vector and $\mathbf{A}'$ denoting the transpose of matrix $\mathbf{A}$, a joint sampling model would take the form

$$\mathbf{x}^l \sim \mathcal{N}\left(\mu^l, \left(\mathbf{\Lambda}^l\right)^{-1}\right), \quad l \in \{1, \cdots, L\}, \quad (4)$$

where

$$\mu^l = \begin{bmatrix} \mu_t^l \\ \mu_s^l \end{bmatrix}, \quad \mathbf{\Lambda}^l = \begin{bmatrix} \mathbf{\Lambda}_t^l & \mathbf{\Lambda}_{ts}^l \\ \mathbf{\Lambda}_{ts}^{l'} & \mathbf{\Lambda}_s^l \end{bmatrix}, \quad (5)$$

where μ^l is the $2d \times 1$ mean vector, and $\mathbf{\Lambda}^l$ is the $2d \times 2d$ precision matrix. $\mathbf{\Lambda}_t^l$ and $\mathbf{\Lambda}_s^l$ account for the interactions of features within the source and target domains, respectively, and $\mathbf{\Lambda}_{ts}^l$ accounts for the interactions of the features across the source and target domains, for any class $l \in \{1, \cdots, L\}$. The key point here is that in transfer learning joint sampling of the source and target domains is not possible, and we always assume that there are two datasets separately sampled from the source and target domains. That is why we use (1) instead of (4) for the sampling model. If we would use (4) as the sampling model, then we would use a Wishart distribution as a prior for the precision matrix $\mathbf{\Lambda}^l$ in (5). This is where we get the idea of defining a joint prior distribution for $\mathbf{\Lambda}_s^l$ and $\mathbf{\Lambda}_t^l$. If $\mathbf{\Lambda}^l$ in (5) has a Wishart distribution, by marginalizing out the term $\mathbf{\Lambda}_{ts}^l$ one could derive a joint distribution for $\mathbf{\Lambda}_s^l$ and $\mathbf{\Lambda}_t^l$.

We present some theorems and definitions that will be used later in deriving the OBTL classifier.

Definition 1. A random $d \times d$ symmetric positive-definite matrix $\mathbf{\Lambda}$ has a nonsingular Wishart distribution with ν degrees of freedom, $W_d(\mathbf{M}, \nu)$, if $\nu \geq d$ and $\mathbf{M}$ is a $d \times d$ positive-definite matrix ($\mathbf{M} > 0$) and the density is

$$p(\mathbf{\Lambda}) = \left[2^{\frac{\nu d}{2}} \Gamma_d\left(\frac{\nu}{2}\right) |\mathbf{M}|^{\frac{\nu}{2}}\right]^{-1} |\mathbf{\Lambda}|^{\frac{\nu-d-1}{2}} \mathrm{etr}\left(-\frac{1}{2}\mathbf{M}^{-1}\mathbf{\Lambda}\right), \quad (6)$$

where $|\mathbf{A}|$ is the determinant of $\mathbf{A}$, $\mathrm{etr}(\mathbf{A}) = \exp\left(\mathrm{tr}(\mathbf{A})\right)$ and $\Gamma_d(\alpha)$ is the multivariate gamma function given by

$$\Gamma_d(\alpha) = \pi^{\frac{d(d-1)}{4}} \prod_{i=1}^{d} \Gamma\left(\alpha - \frac{i-1}{2}\right). \quad (7)$$

Theorem 1. *[25]: If $\mathbf{\Lambda} \sim W_d(\mathbf{M}, \nu)$, and $\mathbf{A}$ is an $r \times d$ matrix of rank r, where $r \leq d$, then $\mathbf{A}\mathbf{\Lambda}\mathbf{A}' \sim W_r(\mathbf{A}\mathbf{M}\mathbf{A}', \nu)$.*

Corollary 1. *If $\mathbf{\Lambda} \sim W_d(\mathbf{M}, \nu)$ and $\mathbf{\Lambda} = \begin{pmatrix} \mathbf{\Lambda}_{11} & \mathbf{\Lambda}_{12} \\ \mathbf{\Lambda}_{12}' & \mathbf{\Lambda}_{22} \end{pmatrix}$, where $\mathbf{\Lambda}_{11}$ and $\mathbf{\Lambda}_{22}$ are $d_1 \times d_1$ and $d_2 \times d_2$ submatrices, respectively, and if $\mathbf{M} = \begin{pmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{12}' & \mathbf{M}_{22} \end{pmatrix}$ is the corresponding partition of $\mathbf{M}$ with $\mathbf{M}_{11}$ and $\mathbf{M}_{22}$ being two $d_1 \times d_1$ and $d_2 \times d_2$ submatrices, respectively, then $\mathbf{\Lambda}_{11} \sim W_{d_1}(\mathbf{M}_{11}, \nu)$ and $\mathbf{\Lambda}_{22} \sim W_{d_2}(\mathbf{M}_{22}, \nu)$.*

Using Corollary 1, we can ensure that using the Wishart distribution for the precision matrix $\mathbf{\Lambda}^l$ (5) of the joint model in (4) will lead to the Wishart marginal distributions for $\mathbf{\Lambda}_s^l$ and $\mathbf{\Lambda}_t^l$ in the source and target domains separately, which is a desired property. Now we introduce a theorem, proposed in [26], which gives the form of the joint distribution of the two submatrices of a partitioned Wishart matrix.

Theorem 2. *[26]: Let $\mathbf{\Lambda} = \begin{pmatrix} \mathbf{\Lambda}_{11} & \mathbf{\Lambda}_{12} \\ \mathbf{\Lambda}_{12}' & \mathbf{\Lambda}_{22} \end{pmatrix}$ be a $(d_1+d_2) \times (d_1+d_2)$ partitioned Wishart random matrix, where the diagonal partitions are of sizes $d_1 \times d_1$ and $d_2 \times d_2$, respectively. The Wishart distribution of $\mathbf{\Lambda}$ has $\nu \geq d_1 + d_2$ degrees of freedom and positive-definite scale matrix $\mathbf{M} = \begin{pmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{12}' & \mathbf{M}_{22} \end{pmatrix}$ partitioned in the same way as $\mathbf{\Lambda}$. The joint distribution of the two diagonal partitions $\mathbf{\Lambda}_{11}$ and $\mathbf{\Lambda}_{22}$ have the density function given by*

$$\begin{aligned} &p(\mathbf{\Lambda}_{11}, \mathbf{\Lambda}_{22}) = \\ &K \, \mathrm{etr}\left(-\frac{1}{2}\left(\mathbf{M}_{11}^{-1} + \mathbf{F}'\mathbf{C}_2\mathbf{F}\right)\mathbf{\Lambda}_{11}\right) \mathrm{etr}\left(-\frac{1}{2}\mathbf{C}_2^{-1}\mathbf{\Lambda}_{22}\right) \\ &\times |\mathbf{\Lambda}_{11}|^{\frac{\nu-d_2-1}{2}} |\mathbf{\Lambda}_{22}|^{\frac{\nu-d_1-1}{2}} \, {}_0F_1\left(\frac{\nu}{2}; \frac{1}{4}\mathbf{G}\right), \end{aligned} \quad (8)$$

where $\mathbf{C}_2 = \mathbf{M}_{22} - \mathbf{M}_{12}'\mathbf{M}_{11}^{-1}\mathbf{M}_{12}$, $\mathbf{F} = \mathbf{C}_2^{-1}\mathbf{M}_{12}'\mathbf{M}_{11}^{-1}$, $\mathbf{G} = \mathbf{\Lambda}_{22}^{\frac{1}{2}}\mathbf{F}\mathbf{\Lambda}_{11}\mathbf{F}'\mathbf{\Lambda}_{22}^{\frac{1}{2}}$, $K^{-1} = 2^{\frac{(d_1+d_2)\nu}{2}}\Gamma_{d_1}\left(\frac{\nu}{2}\right)\Gamma_{d_2}\left(\frac{\nu}{2}\right)|\mathbf{M}|^{\frac{\nu}{2}}$, and ${}_0F_1$ is the generalized matrix-variate hypergeometric function.

Definition 2. [27]: The generalized hypergeometric function of one matrix argument is defined by

$$\begin{aligned} &{}_pF_q(a_1, \cdots, a_p; b_1, \cdots, b_q; \mathbf{X}) \\ &= \sum_{k=0}^{\infty} \sum_{\kappa \vdash k} \frac{(a_1)_\kappa \cdots (a_p)_\kappa}{(b_1)_\kappa \cdots (b_q)_\kappa} \frac{C_\kappa(\mathbf{X})}{k!}, \quad (9) \end{aligned}$$

where a_i, $i = 1, \cdots, p$, and b_j, $j = 1, \cdots, q$, are arbitrary complex (real in our case) numbers, $C_\kappa(\mathbf{X})$ is the zonal polynomial of $d \times d$ symmetric matrix $\mathbf{X}$ corresponding to the ordered partition $\kappa = (k_1, \cdots, k_d)$, $k_1 \geq \cdots \geq k_d \geq 0$,

$k_1 + \cdots k_d = k$ and $\sum_{\kappa \vdash k}$ denotes summation over all partitions κ of k. The generalized hypergeometric coefficient $(a)_\kappa$ is defined by

$$(a)_\kappa = \prod_{i=1}^{d} \left(a - \frac{i-1}{2} \right)_{k_i}, \tag{10}$$

where $(a)_r = a(a+1)\cdots(a+r-1)$, $r = 1, 2, \cdots$, with $(a)_0 = 1$.

Conditions for convergence of the series in (9) are available in the literature [28]. From (9) it follows

$$_0F_0(\mathbf{X}) = \sum_{k=0}^{\infty} \sum_{\kappa \vdash k} \frac{C_\kappa(\mathbf{X})}{k!} = \sum_{k=0}^{\infty} \frac{(\mathrm{tr}(\mathbf{X}))^k}{k!} = \mathrm{etr}(\mathbf{X}),$$

$$_1F_0(a; \mathbf{X}) = \sum_{k=0}^{\infty} \sum_{\kappa \vdash k} \frac{(a)_\kappa C_\kappa(\mathbf{X})}{k!} = |\mathbf{I}_m - \mathbf{X}|^{-a}, \quad ||\mathbf{X}|| < 1,$$

$$_0F_1(b; \mathbf{X}) = \sum_{k=0}^{\infty} \sum_{\kappa \vdash k} \frac{C_\kappa(\mathbf{X})}{(b)_\kappa k!},$$

$$_1F_1(a; b; \mathbf{X}) = \sum_{k=0}^{\infty} \sum_{\kappa \vdash k} \frac{(a)_\kappa}{(b)_\kappa} \frac{C_\kappa(\mathbf{X})}{k!},$$

$$_2F_1(a, b; c; \mathbf{X}) = \sum_{k=0}^{\infty} \sum_{\kappa \vdash k} \frac{(a)_\kappa (b)_\kappa}{(c)_\kappa} \frac{C_\kappa(\mathbf{X})}{k!}, \quad ||\mathbf{X}|| < 1, \tag{11}$$

where $||\mathbf{X}|| < 1$ means that the maximum of the absolute values of the eigenvalues of $\mathbf{X}$ is less than 1. $_1F_1(a; b; \mathbf{X})$ and $_2F_1(a, b; c; \mathbf{X})$ are respectively called Confluent and Gauss hypergeometric functions of matrix argument.

Theorem 3. *[25]: Let $\mathbf{Z}$ be a complex symmetric matrix whose real part is positive-definite, and let $\mathbf{X}$ be an arbitrary complex symmetric matrix. Then*

$$\int_{\mathbf{R}>0} \mathrm{etr}(-\mathbf{Z}\mathbf{R})|\mathbf{R}|^{\alpha - \frac{d+1}{2}} C_\kappa(\mathbf{R}\mathbf{X}) d\mathbf{R} = \Gamma_d(\alpha)(\alpha)_\kappa |\mathbf{Z}|^{-\alpha} C_\kappa(\mathbf{X}\mathbf{Z}^{-1}), \tag{12}$$

the integration being over the space of positive-definite $d \times d$ matrices, and valid for all complex numbers α satisfying $\mathrm{Re}(\alpha) > \frac{d-1}{2}$. $\Gamma_d(\alpha)$ is the multivariate gamma function defined in (7).

Theorem 4. *[29]: The zonal polynomials are invariant under orthogonal transformation. That is, for a $d \times d$ symmetric matrix $\mathbf{X}$,*

$$C_\kappa(\mathbf{X}) = C_\kappa(\mathbf{H}\mathbf{X}\mathbf{H}'), \tag{13}$$

where $\mathbf{H}$ is an orthogonal matrix of order d. If $\mathbf{R}$ is a symmetric positive-definite matrix of order d, then

$$C_\kappa(\mathbf{R}\mathbf{X}) = C_\kappa(\mathbf{R}^{1/2}\mathbf{X}\mathbf{R}^{1/2}). \tag{14}$$

As a result, if $\mathbf{R}$ is a symmetric positive-definite matrix, the hypergeometric function has the following property:

$$_pF_q(a_1, \cdots, a_p; b_1, \cdots, b_q; \mathbf{R}\mathbf{X}) = {}_pF_q(a_1, \cdots, a_p; b_1, \cdots, b_q; \mathbf{R}^{1/2}\mathbf{X}\mathbf{R}^{1/2}). \tag{15}$$

Theorem 5. *[30]: If $\mathbf{Z} > 0$ and $\mathrm{Re}(\alpha) > \frac{d-1}{2}$, and $\mathbf{X}$ is a*

$d \times d$ *symmetric matrix, we have*

$$\int_{\mathbf{R}>0} \mathrm{etr}(-\mathbf{Z}\mathbf{R})|\mathbf{R}|^{\alpha - \frac{d+1}{2}}$$
$$\times {}_pF_q(a_1, \cdots, a_p; b_1, \cdots, b_q; \mathbf{R}\mathbf{X}) d\mathbf{R}$$
$$= \int_{\mathbf{R}>0} \mathrm{etr}(-\mathbf{Z}\mathbf{R})|\mathbf{R}|^{\alpha - \frac{d+1}{2}}$$
$$\times {}_pF_q(a_1, \cdots, a_p; b_1, \cdots, b_q; \mathbf{R}^{1/2}\mathbf{X}\mathbf{R}^{1/2}) d\mathbf{R}$$
$$= \Gamma_d(\alpha)|\mathbf{Z}|^{-\alpha} {}_{p+1}F_q(a_1, \cdots, a_p, \alpha; b_1, \cdots, b_q; \mathbf{X}\mathbf{Z}^{-1}).$$

Now, using Theorem 2, we define the joint prior distribution, $p(\mathbf{\Lambda}_s^l, \mathbf{\Lambda}_t^l)$ in (2), of the precision matrices of the source and target domains for class $l \in \{1, \cdots, L\}$ as follows:

$$p(\mathbf{\Lambda}_t^l, \mathbf{\Lambda}_s^l) = K^l \mathrm{etr}\left(-\frac{1}{2}\left((\mathbf{M}_t^l)^{-1} + \mathbf{F}^{l'}\mathbf{C}^l\mathbf{F}^l \right) \mathbf{\Lambda}_t^l \right)$$
$$\times \mathrm{etr}\left(-\frac{1}{2}(\mathbf{C}^l)^{-1}\mathbf{\Lambda}_s^l \right) \tag{16}$$
$$\times \left| \mathbf{\Lambda}_t^l \right|^{\frac{\nu^l - d - 1}{2}} \left| \mathbf{\Lambda}_s^l \right|^{\frac{\nu^l - d - 1}{2}} {}_0F_1\left(\frac{\nu^l}{2}; \frac{1}{4}\mathbf{G}^l \right),$$

where $\mathbf{M}^l = \begin{pmatrix} \mathbf{M}_t^l, & \mathbf{M}_{ts}^l \\ \mathbf{M}_{ts}^{l'}, & \mathbf{M}_s^l \end{pmatrix}$ is a $2d \times 2d$ positive definite scale matrix, $\nu^l \geq 2d$ denotes degrees of freedom, $\mathbf{C}^l = \mathbf{M}_s^l - \mathbf{M}_{ts}^{l'}(\mathbf{M}_t^l)^{-1}\mathbf{M}_{ts}^l$, $\mathbf{F}^l = (\mathbf{C}^l)^{-1}\mathbf{M}_{ts}^{l'}(\mathbf{M}_t^l)^{-1}$, $\mathbf{G}^l = \mathbf{\Lambda}_s^{l\frac{1}{2}}\mathbf{F}^l\mathbf{\Lambda}_t^l\mathbf{F}^{l'}\mathbf{\Lambda}_s^{l\frac{1}{2}}$, and $(K^l)^{-1} = 2^{d\nu^l}\Gamma_d^2\left(\frac{\nu^l}{2}\right)|\mathbf{M}^l|^{\frac{\nu^l}{2}}$. Using Corollary 1, $\mathbf{\Lambda}_t^l$ and $\mathbf{\Lambda}_s^l$ have the following Wishart marginal distributions:

$$\mathbf{\Lambda}_z^l \sim W_d(\mathbf{M}_z^l, \nu^l), \quad l \in \{1, \cdots, L\}, \quad z \in \{s, t\}. \tag{17}$$

III. POSTERIORS OF TARGET PARAMETERS

Having defined the prior distributions in the previous section, we aim to derive the posterior distribution of the parameters of the target domain upon observing the training source $\mathcal{D}_s$ and target $\mathcal{D}_t$ datasets. The likelihood of the datasets $\mathcal{D}_t$ and $\mathcal{D}_s$ is conditionally independent given the parameters of the target and source domains. The dependence between the two domains is due to the dependence of the prior distributions of the precision matrices, as shown in Fig 1. Within each domain, source or target, the likelihoods of the different classes are also conditionally independent given the parameters of the classes. As such, the joint likelihood of the datasets $\mathcal{D}_t$ and $\mathcal{D}_s$ can be written as

$$p(\mathcal{D}_t, \mathcal{D}_s | \mu_t, \mu_s, \mathbf{\Lambda}_t, \mathbf{\Lambda}_s) = p(\mathcal{D}_t | \mu_t, \mathbf{\Lambda}_t) p(\mathcal{D}_s | \mu_s, \mathbf{\Lambda}_s)$$
$$= p(\mathcal{D}_t^1, \cdots, \mathcal{D}_t^L | \mu_t^1, \cdots, \mu_t^L, \mathbf{\Lambda}_t^1, \cdots, \mathbf{\Lambda}_t^L)$$
$$\times p(\mathcal{D}_s^1, \cdots, \mathcal{D}_s^L | \mu_s^1, \cdots, \mu_s^L, \mathbf{\Lambda}_s^1, \cdots, \mathbf{\Lambda}_s^L) \tag{18}$$
$$= \prod_{l=1}^{L} p(\mathcal{D}_t^l | \mu_t^l, \mathbf{\Lambda}_t^l) \prod_{l=1}^{L} p(\mathcal{D}_s^l | \mu_s^l, \mathbf{\Lambda}_s^l).$$

The posterior of the parameters given $\mathcal{D}_t$ and $\mathcal{D}_s$ satisfies

$$p(\mu_t, \mu_s, \mathbf{\Lambda}_t, \mathbf{\Lambda}_s | \mathcal{D}_t, \mathcal{D}_s)$$
$$\propto p(\mathcal{D}_t, \mathcal{D}_s | \mu_t, \mu_s, \mathbf{\Lambda}_t, \mathbf{\Lambda}_s) p(\mu_t, \mu_s, \mathbf{\Lambda}_t, \mathbf{\Lambda}_s)$$
$$\propto \prod_{l=1}^{L} p(\mathcal{D}_t^l | \mu_t^l, \mathbf{\Lambda}_t^l) \prod_{l=1}^{L} p(\mathcal{D}_s^l | \mu_s^l, \mathbf{\Lambda}_s^l) \prod_{l=1}^{L} p(\mu_t^l, \mu_s^l, \mathbf{\Lambda}_t^l, \mathbf{\Lambda}_s^l), \tag{19}$$
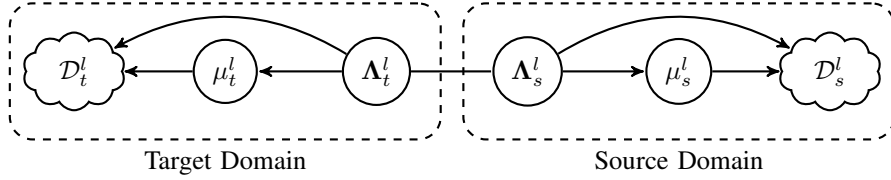
Figure 1: Dependency of the source and target domains through their precision matrices for any class $l \in \{1, \cdots, L\}$.

where we assume that the priors of the parameters in different classes are independent, $p(\mu_t, \mu_s, \Lambda_t, \Lambda_s) = \prod_{l=1}^{L} p(\mu_t^l, \mu_s^l, \Lambda_t^l, \Lambda_s^l)$. From (2) and (19),

$$p(\mu_t, \mu_s, \Lambda_t, \Lambda_s | \mathcal{D}_t, \mathcal{D}_s) \propto \prod_{l=1}^{L} p(\mathcal{D}_t^l | \mu_t^l, \Lambda_t^l) p(\mathcal{D}_s^l | \mu_s^l, \Lambda_s^l)$$
$$\times p\left(\mu_s^l | \Lambda_s^l\right) p\left(\mu_t^l | \Lambda_t^l\right) p\left(\Lambda_s^l, \Lambda_t^l\right). \tag{20}$$

We can see that the posterior of the parameters is equal to the product of the posteriors of the parameters of each class:

$$p(\mu_t, \mu_s, \Lambda_t, \Lambda_s | \mathcal{D}_t, \mathcal{D}_s) = \prod_{l=1}^{L} p(\mu_t^l, \mu_s^l, \Lambda_t^l, \Lambda_s^l | \mathcal{D}_t^l, \mathcal{D}_s^l), \tag{21}$$

where

$$p(\mu_t^l, \mu_s^l, \Lambda_t^l, \Lambda_s^l | \mathcal{D}_t^l, \mathcal{D}_s^l) \propto p(\mathcal{D}_t^l | \mu_t^l, \Lambda_t^l) p(\mathcal{D}_s^l | \mu_s^l, \Lambda_s^l)$$
$$\times p\left(\mu_s^l | \Lambda_s^l\right) p\left(\mu_t^l | \Lambda_t^l\right) p\left(\Lambda_s^l, \Lambda_t^l\right). \tag{22}$$

Since we are interested in the posterior of the parameters of the target domain, we integrate out the parameters of the source domain in (21):

$$p(\mu_t, \Lambda_t | \mathcal{D}_t, \mathcal{D}_s) = \int_{\mu_s, \Lambda_s} p(\mu_t, \mu_s, \Lambda_t, \Lambda_s | \mathcal{D}_t, \mathcal{D}_s) d\mu_s d\Lambda_s$$
$$= \prod_{l=1}^{L} \int_{\mu_s^l, \Lambda_s^l} p(\mu_t^l, \mu_s^l, \Lambda_t^l, \Lambda_s^l | \mathcal{D}_t^l, \mathcal{D}_s^l) d\mu_s^l d\Lambda_s^l$$
$$= \prod_{l=1}^{L} p(\mu_t^l, \Lambda_t^l | \mathcal{D}_t^l, \mathcal{D}_s^l),$$

where

$$p(\mu_t^l, \Lambda_t^l | \mathcal{D}_t^l, \mathcal{D}_s^l)$$
$$= \int_{\mu_s^l, \Lambda_s^l} p(\mu_t^l, \mu_s^l, \Lambda_t^l, \Lambda_s^l | \mathcal{D}_t^l, \mathcal{D}_s^l) d\mu_s^l d\Lambda_s^l$$
$$\propto p(\mathcal{D}_t^l | \mu_t^l, \Lambda_t^l) p\left(\mu_t^l | \Lambda_t^l\right)$$
$$\times \int_{\mu_s^l, \Lambda_s^l} p(\mathcal{D}_s^l | \mu_s^l, \Lambda_s^l) p\left(\mu_s^l | \Lambda_s^l\right) p\left(\Lambda_s^l, \Lambda_t^l\right) d\mu_s^l d\Lambda_s^l. \tag{23}$$

From (1), for each domain $z \in \{s, t\}$,

$$p(\mathcal{D}_z^l | \mu_z^l, \Lambda_z^l) = (2\pi)^{-\frac{d n_z^l}{2}} \left|\Lambda_z^l\right|^{\frac{n_z^l}{2}} \exp\left(-\frac{1}{2} \mathbf{Q}_z^l\right), \tag{24}$$

where $\mathbf{Q}_z^l = \sum_{i=1}^{n_z^l} \left(\mathbf{x}_{z,i}^l - \mu_z^l\right)' \Lambda_z^l \left(\mathbf{x}_{z,i}^l - \mu_z^l\right)$. Moreover, from (3), for each domain $z \in \{s, t\}$,

$$p\left(\mu_z^l | \Lambda_z^l\right) = (2\pi)^{-\frac{d}{2}} \left(\kappa_z^l\right)^{\frac{d}{2}} \left|\Lambda_z^l\right|^{\frac{1}{2}}$$
$$\times \exp\left(-\frac{\kappa_z^l}{2} \left(\mu_z^l - \mathbf{m}_z^l\right)' \Lambda_z^l \left(\mu_z^l - \mathbf{m}_z^l\right)\right). \tag{25}$$

From (16), (23), (24), and (25),

$$p(\mu_t^l, \Lambda_t^l | \mathcal{D}_t^l, \mathcal{D}_s^l) \propto \left|\Lambda_t^l\right|^{\frac{n_t^l}{2}} \exp\left(-\frac{1}{2} \mathbf{Q}_t^l\right) \left|\Lambda_t^l\right|^{\frac{1}{2}}$$

$$\times \exp\left(-\frac{\kappa_t^l}{2} \left(\mu_t^l - \mathbf{m}_t^l\right)' \Lambda_t^l \left(\mu_t^l - \mathbf{m}_t^l\right)\right)$$

$$\times \left|\Lambda_t^l\right|^{\frac{\nu^l - d - 1}{2}} \text{etr}\left(-\frac{1}{2} \left(\left(\mathbf{M}_t^l\right)^{-1} + \mathbf{F}^{l'} \mathbf{C}^l \mathbf{F}^l\right) \Lambda_t^l\right)$$

$$\times \int_{\mu_s^l, \Lambda_s^l} \left\{\left|\Lambda_s^l\right|^{\frac{n_s^l}{2}} \exp\left(-\frac{1}{2} \mathbf{Q}_s^l\right) \left|\Lambda_s^l\right|^{\frac{1}{2}}\right. \tag{26}$$

$$\times \exp\left(-\frac{\kappa_s^l}{2} \left(\mu_s^l - \mathbf{m}_s^l\right)' \Lambda_s^l \left(\mu_s^l - \mathbf{m}_s^l\right)\right)$$

$$\times \left|\Lambda_s^l\right|^{\frac{\nu^l - d - 1}{2}} \text{etr}\left(-\frac{1}{2} \left(\mathbf{C}^l\right)^{-1} \Lambda_s^l\right)$$

$$\times \left. {}_0F_1\left(\frac{\nu^l}{2}; \frac{1}{4} \Lambda_s^{l\,\frac{1}{2}} \mathbf{F}^l \Lambda_t^l \mathbf{F}^{l'} \Lambda_s^{l\,\frac{1}{2}}\right)\right\} d\mu_s^l d\Lambda_s^l.$$

Lemma 1. *If $\mathcal{D} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ where $\mathbf{x}_i$ is a $d \times 1$ vector and $\mathbf{x}_i \sim \mathcal{N}(\mu, (\Lambda)^{-1})$, for $i = 1, \cdots, n$, and (μ, Λ) has a Normal-Wishart prior, such that, $\mu | \Lambda \sim \mathcal{N}(\mathbf{m}, (\kappa\Lambda)^{-1})$ and $\Lambda \sim W_d(\mathbf{M}, \nu)$, then the posterior of (μ, Λ) upon observing $\mathcal{D}$ is also a Normal-Wishart distribution:*

$$\mu | \Lambda, \mathcal{D} \sim \mathcal{N}(\mathbf{m}_n, (\kappa_n \Lambda)^{-1}),$$
$$\Lambda | \mathcal{D} \sim W_d(\mathbf{M}_n, \nu_n), \tag{27}$$

where

$$\kappa_n = \kappa + n, \quad \nu_n = \nu + n, \quad \mathbf{m}_n = \frac{\kappa\mathbf{m} + n\bar{\mathbf{x}}}{\kappa + n},$$
$$\mathbf{M}_n^{-1} = \mathbf{M}^{-1} + \mathbf{S} + \frac{\kappa n}{\kappa + n}(\mathbf{m} - \bar{\mathbf{x}})(\mathbf{m} - \bar{\mathbf{x}})', \tag{28}$$

depending on the sample mean and covariance matrix

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i, \qquad \mathbf{S} = \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})'. \tag{29}$$

Using Lemma 1 we can simplify (26) as

$$p(\mu_t^l, \Lambda_t^l | \mathcal{D}_t^l, \mathcal{D}_s^l)$$

$$\propto \left|\Lambda_t^l\right|^{\frac{1}{2}} \exp\left(-\frac{\kappa_{t,n}^l}{2} \left(\mu_t^l - \mathbf{m}_{t,n}^l\right)' \Lambda_t^l \left(\mu_t^l - \mathbf{m}_{t,n}^l\right)\right)$$

$$\times \left|\Lambda_t^l\right|^{\frac{\nu^l + n_t^l - d - 1}{2}} \text{etr}\left(-\frac{1}{2} \left(\mathbf{T}_t^l\right)^{-1} \Lambda_t^l\right)$$

$$\int_{\mu_s^l, \Lambda_s^l} \left\{\left|\Lambda_s^l\right|^{\frac{1}{2}} \exp\left(-\frac{\kappa_{s,n}^l}{2} \left(\mu_s^l - \mathbf{m}_{s,n}^l\right)' \Lambda_s^l \left(\mu_s^l - \mathbf{m}_{s,n}^l\right)\right)\right.$$

$$\times \left|\Lambda_s^l\right|^{\frac{\nu^l + n_s^l - d - 1}{2}} \text{etr}\left(-\frac{1}{2} \left(\mathbf{T}_s^l\right)^{-1} \Lambda_s^l\right)$$

$$\times \left. {}_0F_1\left(\frac{\nu^l}{2}; \frac{1}{4} \Lambda_s^{l\,\frac{1}{2}} \mathbf{F}^l \Lambda_t^l \mathbf{F}^{l'} \Lambda_s^{l\,\frac{1}{2}}\right)\right\} d\mu_s^l d\Lambda_s^l, \tag{30}$$

where

$$\kappa_{t,n}^l = \kappa_t^l + n_t^l, \qquad \kappa_{s,n}^l = \kappa_s^l + n_s^l,$$

$$\mathbf{m}_{t,n}^l = \frac{\kappa_t^l \mathbf{m}_t^l + n_t^l \bar{\mathbf{x}}_t^l}{\kappa_t^l + n_t^l}, \qquad \mathbf{m}_{s,n}^l = \frac{\kappa_s^l \mathbf{m}_s^l + n_s^l \bar{\mathbf{x}}_s^l}{\kappa_s^l + n_s^l},$$

$$\left(\mathbf{T}_t^l\right)^{-1} = \left(\mathbf{M}_t^l\right)^{-1} + \mathbf{F}^{l'} \mathbf{C}^l \mathbf{F}^l + \mathbf{S}_t^l$$
$$+ \frac{\kappa_t^l n_t^l}{\kappa_t^l + n_t^l}(\mathbf{m}_t^l - \bar{\mathbf{x}}_t^l)(\mathbf{m}_t^l - \bar{\mathbf{x}}_t^l)',$$

$$\left(\mathbf{T}_s^l\right)^{-1} = \left(\mathbf{C}^l\right)^{-1} + \mathbf{S}_s^l + \frac{\kappa_s^l n_s^l}{\kappa_s^l + n_s^l}(\mathbf{m}_s^l - \bar{\mathbf{x}}_s^l)(\mathbf{m}_s^l - \bar{\mathbf{x}}_s^l)',$$
(31)

with sample means and covariances for $z \in \{s, t\}$ as

$$\bar{\mathbf{x}}_z^l = \frac{1}{n_z^l} \sum_{i=1}^{n_z^l} \mathbf{x}_{z,i}^l, \quad \mathbf{S}_z^l = \sum_{i=1}^{n_z^l} \left(\mathbf{x}_{z,i}^l - \bar{\mathbf{x}}_z^l\right)\left(\mathbf{x}_{z,i}^l - \bar{\mathbf{x}}_z^l\right)'.$$

Using the equation

$$\int_{\mathbf{x}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)' \mathbf{\Lambda}(\mathbf{x} - \mu)\right) d\mathbf{x} = (2\pi)^{\frac{d}{2}} |\mathbf{\Lambda}|^{-\frac{1}{2}}, \quad (32)$$

and integrating out μ_s^l in (30) yields

$$p(\mu_t^l, \mathbf{\Lambda}_t^l | \mathcal{D}_t^l, \mathcal{D}_s^l)$$
$$\propto |\mathbf{\Lambda}_t^l|^{\frac{1}{2}} \exp\left(-\frac{\kappa_{t,n}^l}{2}\left(\mu_t^l - \mathbf{m}_{t,n}^l\right)' \mathbf{\Lambda}_t^l \left(\mu_t^l - \mathbf{m}_{t,n}^l\right)\right)$$
$$\times |\mathbf{\Lambda}_t^l|^{\frac{\nu^l + n_t^l - d - 1}{2}} \text{etr}\left(-\frac{1}{2}\left(\mathbf{T}_t^l\right)^{-1} \mathbf{\Lambda}_t^l\right)$$
$$\times \int_{\mathbf{\Lambda}_s^l} \left\{ |\mathbf{\Lambda}_s^l|^{\frac{\nu^l + n_s^l - d - 1}{2}} \text{etr}\left(-\frac{1}{2}\left(\mathbf{T}_s^l\right)^{-1} \mathbf{\Lambda}_s^l\right) \right.$$
$$\left. \times {}_0F_1\left(\frac{\nu^l}{2}; \frac{1}{4}\mathbf{\Lambda}_s^{l\frac{1}{2}} \mathbf{F}^l \mathbf{\Lambda}_t^l \mathbf{F}^{l'} \mathbf{\Lambda}_s^{l\frac{1}{2}}\right)\right\} d\mathbf{\Lambda}_s^l.$$
(33)

The integral, I, in (33) can be done using Theorem 5 as

$$I = \Gamma_d\left(\frac{\nu^l + n_s^l}{2}\right)$$
$$\times \left|2\mathbf{T}_s^l\right|^{\frac{\nu^l + n_s^l}{2}} {}_1F_1\left(\frac{\nu^l + n_s^l}{2}; \frac{\nu^l}{2}; \frac{1}{2}\mathbf{F}^l \mathbf{\Lambda}_t^l \mathbf{F}^{l'} \mathbf{T}_s^l\right),$$
(34)

where ${}_1F_1(a; b; \mathbf{X})$ is the Confluent hypergeometric function with the matrix argument $\mathbf{X}$. As a result, (33) becomes

$$p(\mu_t^l, \mathbf{\Lambda}_t^l | \mathcal{D}_t^l, \mathcal{D}_s^l) =$$
$$A^l |\mathbf{\Lambda}_t^l|^{\frac{1}{2}} \exp\left(-\frac{\kappa_{t,n}^l}{2}\left(\mu_t^l - \mathbf{m}_{t,n}^l\right)' \mathbf{\Lambda}_t^l \left(\mu_t^l - \mathbf{m}_{t,n}^l\right)\right)$$
$$\times |\mathbf{\Lambda}_t^l|^{\frac{\nu^l + n_t^l - d - 1}{2}} \text{etr}\left(-\frac{1}{2}\left(\mathbf{T}_t^l\right)^{-1} \mathbf{\Lambda}_t^l\right)$$
$$\times {}_1F_1\left(\frac{\nu^l + n_s^l}{2}; \frac{\nu^l}{2}; \frac{1}{2}\mathbf{F}^l \mathbf{\Lambda}_t^l \mathbf{F}^{l'} \mathbf{T}_s^l\right).$$
(35)

where the constant of proportionality, A^l, makes the integration of the posterior $p(\mu_t^l, \mathbf{\Lambda}_t^l | \mathcal{D}_t^l, \mathcal{D}_s^l)$ with respect to μ_t^l and $\mathbf{\Lambda}_t^l$ equal to one. Hence,

$$\left(A^l\right)^{-1} = \int_{\mathbf{\Lambda}_t^l} |\mathbf{\Lambda}_t^l|^{\frac{\nu^l + n_t^l - d - 1}{2}} \text{etr}\left(-\frac{1}{2}\left(\mathbf{T}_t^l\right)^{-1} \mathbf{\Lambda}_t^l\right) |\mathbf{\Lambda}_t^l|^{\frac{1}{2}}$$

$$\times \int_{\mu_t^l} \exp\left(-\frac{\kappa_{t,n}^l}{2}\left(\mu_t^l - \mathbf{m}_{t,n}^l\right)' \mathbf{\Lambda}_t^l \left(\mu_t^l - \mathbf{m}_{t,n}^l\right)\right) d\mu_t^l$$

$$\times {}_1F_1\left(\frac{\nu^l + n_s^l}{2}; \frac{\nu^l}{2}; \frac{1}{2}\mathbf{F}^l \mathbf{\Lambda}_t^l \mathbf{F}^{l'} \mathbf{T}_s^l\right) d\mathbf{\Lambda}_t^l.$$
(36)

Using (32), the inner integral equals to $(2\pi)^{\frac{d}{2}} |\kappa_{t,n}^l \mathbf{\Lambda}_t^l|^{-\frac{1}{2}} = \left(\frac{2\pi}{\kappa_{t,n}^l}\right)^{\frac{d}{2}} |\mathbf{\Lambda}_t^l|^{-\frac{1}{2}}$. Hence,

$$\left(A^l\right)^{-1} = \left(\frac{2\pi}{\kappa_{t,n}^l}\right)^{\frac{d}{2}} \int_{\mathbf{\Lambda}_t^l} |\mathbf{\Lambda}_t^l|^{\frac{\nu^l + n_t^l - d - 1}{2}} \text{etr}\left(-\frac{1}{2}\left(\mathbf{T}_t^l\right)^{-1} \mathbf{\Lambda}_t^l\right)$$
$$\times {}_1F_1\left(\frac{\nu^l + n_s^l}{2}; \frac{\nu^l}{2}; \frac{1}{2}\mathbf{F}^l \mathbf{\Lambda}_t^l \mathbf{F}^{l'} \mathbf{T}_s^l\right) d\mathbf{\Lambda}_t^l.$$
(37)

Letting the variable change $\Omega = \mathbf{F}^l \mathbf{\Lambda}_t^l \mathbf{F}^{l'}$, we have $d\Omega = |\mathbf{F}^l|^{d+1} d\mathbf{\Lambda}_t^l$ and $\mathbf{\Lambda}_t^l = \left(\mathbf{F}^l\right)^{-1} \Omega \left(\mathbf{F}^{l'}\right)^{-1}$. Since $\text{tr}(\mathbf{ABCD}) = \text{tr}(\mathbf{BCDA}) = \text{tr}(\mathbf{CDAB}) = \text{tr}(\mathbf{DABC})$ and $|\mathbf{ABC}| = |\mathbf{A}||\mathbf{B}||\mathbf{C}|$, A^l can be derived as

$$\left(A^l\right)^{-1} = \left(\frac{2\pi}{\kappa_{t,n}^l}\right)^{\frac{d}{2}} |\mathbf{F}^l|^{-(\nu^l + n_t^l)} \int_{\Omega} \left\{ |\Omega|^{\frac{\nu^l + n_t^l - d - 1}{2}} \right.$$
$$\times \text{etr}\left(-\frac{1}{2}\left(\mathbf{F}^{l'}\right)^{-1}\left(\mathbf{T}_t^l\right)^{-1}\mathbf{F}^{l-1}\Omega\right)$$
$$\left. \times {}_1F_1\left(\frac{\nu^l + n_s^l}{2}; \frac{\nu^l}{2}; \frac{1}{2}\Omega \mathbf{T}_s^l\right)\right\} d\Omega$$
(38)
$$= \left(\frac{2\pi}{\kappa_{t,n}^l}\right)^{\frac{d}{2}} 2^{\frac{d\left(\nu^l + n_t^l\right)}{2}} \Gamma_d\left(\frac{\nu^l + n_t^l}{2}\right) |\mathbf{T}_t^l|^{\frac{\nu^l + n_t^l}{2}}$$
$$\times {}_2F_1\left(\frac{\nu^l + n_s^l}{2}, \frac{\nu^l + n_t^l}{2}; \frac{\nu^l}{2}; \mathbf{T}_s^l \mathbf{F}^l \mathbf{T}_t^l \mathbf{F}^{l'}\right),$$

where the second equality follows from Theorem 5, and ${}_2F_1(a, b; c; \mathbf{X})$ is the Gauss hypergeometric function with the matrix argument $\mathbf{X}$. As such, we have derived the closed-form posterior distribution of the target parameters $(\mu_t^l, \mathbf{\Lambda}_t^l)$ in (35), where A^l is given by (38).

IV. EFFECTIVE CLASS-CONDITIONAL DENSITIES

For optimal Bayesian classifier [23], [24], using the posterior predictive densities of the classes, called "effective class-conditional densities", leads to the optimal choices for classifiers in order to minimize the Bayesian error estimates of the classifiers. Similarly, we can derive the effective class-conditional densities for defining the OBTL classifier in the target domain, albeit with the posterior of the target parameters derived from both the target and source datasets.

Suppose that $\mathbf{x}$ denotes a $d \times 1$ new observed data point in the target domain, where we aim to optimally classify it to one of the classes $l \in \{1, \cdots, L\}$. In the context of the optimal Bayesian classifier, we need the effective class-conditional densities for the L classes, defined as

$$p(\mathbf{x}|l) = \int_{\mu_t^l, \mathbf{\Lambda}_t^l} p(\mathbf{x}|\mu_t^l, \mathbf{\Lambda}_t^l)\pi^\star(\mu_t^l, \mathbf{\Lambda}_t^l)d\mu_t^l d\mathbf{\Lambda}_t^l, \quad (39)$$

for $l \in \{1, \cdots, L\}$, where $\pi^{\star}(\mu_t^l, \mathbf{\Lambda}_t^l) = p(\mu_t^l, \mathbf{\Lambda}_t^l | \mathcal{D}_t^l, \mathcal{D}_s^l)$ is the posterior of $(\mu_t^l, \mathbf{\Lambda}_t^l)$ upon observation of $\mathcal{D}_t^l$ and $\mathcal{D}_s^l$. The likelihood $p(\mathbf{x}|\mu_t^l, \mathbf{\Lambda}_t^l)$ and posterior $p(\mu_t^l, \mathbf{\Lambda}_t^l | \mathcal{D}_t^l, \mathcal{D}_s^l)$ are given in (1) and (35), respectively. Hence,

$$
\begin{aligned}
p(\mathbf{x}|l) = (2\pi)^{-\frac{d}{2}} A^l \int_{\mu_t^l, \mathbf{\Lambda}_t^l} & \left\{ |\mathbf{\Lambda}_t^l|^{\frac{1}{2}} \right. \\
& \times \exp\left(-\frac{1}{2}\left(\mathbf{x} - \mu_t^l\right)' \mathbf{\Lambda}_t^l \left(\mathbf{x} - \mu_t^l\right)\right) \\
& \times |\mathbf{\Lambda}_t^l|^{\frac{1}{2}} \exp\left(-\frac{\kappa_{t,n}^l}{2}\left(\mu_t^l - \mathbf{m}_{t,n}^l\right)' \mathbf{\Lambda}_t^l \left(\mu_t^l - \mathbf{m}_{t,n}^l\right)\right) \\
& \times |\mathbf{\Lambda}_t^l|^{\frac{\nu^l + n_t^l - d - 1}{2}} \operatorname{etr}\left(-\frac{1}{2}\left(\mathbf{T}_t^l\right)^{-1} \mathbf{\Lambda}_t^l\right) \\
& \left. \times {}_1F_1\left(\frac{\nu^l + n_s^l}{2}; \frac{\nu^l}{2}; \frac{1}{2}\mathbf{F}^l \mathbf{\Lambda}_t^l \mathbf{F}^{l'} \mathbf{T}_s^l\right) \right\} d\mu_t^l d\mathbf{\Lambda}_t^l.
\end{aligned}
\tag{40}
$$

Similarly, we can simplify (40) as

$$
\begin{aligned}
p(\mathbf{x}|l) = (2\pi)^{-\frac{d}{2}} A^l \int_{\mu_t^l, \mathbf{\Lambda}_t^l} & \left\{ |\mathbf{\Lambda}_t^l|^{\frac{1}{2}} \right. \\
& \times \exp\left(-\frac{\kappa_{\mathbf{x}}^l}{2}\left(\mu_t^l - \mathbf{m}_{\mathbf{x}}^l\right)' \mathbf{\Lambda}_t^l \left(\mu_t^l - \mathbf{m}_{\mathbf{x}}^l\right)\right) \\
& \times |\mathbf{\Lambda}_t^l|^{\frac{\nu^l + n_t^l + 1 - d - 1}{2}} \operatorname{etr}\left(-\frac{1}{2}\left(\mathbf{T}_{\mathbf{x}}^l\right)^{-1} \mathbf{\Lambda}_t^l\right) \\
& \left. \times {}_1F_1\left(\frac{\nu^l + n_s^l}{2}; \frac{\nu^l}{2}; \frac{1}{2}\mathbf{F}^l \mathbf{\Lambda}_t^l \mathbf{F}^{l'} \mathbf{T}_s^l\right) \right\} d\mu_t^l d\mathbf{\Lambda}_t^l,
\end{aligned}
\tag{41}
$$

where

$$
\kappa_{\mathbf{x}}^l = \kappa_{t,n}^l + 1 = \kappa_t^l + n_t^l + 1, \qquad \mathbf{m}_{\mathbf{x}}^l = \frac{\kappa_{t,n}^l \mathbf{m}_{t,n}^l + \mathbf{x}}{\kappa_{t,n}^l + 1},
$$

$$
\left(\mathbf{T}_{\mathbf{x}}^l\right)^{-1} = \left(\mathbf{T}_t^l\right)^{-1} + \frac{\kappa_{t,n}^l}{\kappa_{t,n}^l + 1}\left(\mathbf{m}_{t,n}^l - \mathbf{x}\right)\left(\mathbf{m}_{t,n}^l - \mathbf{x}\right)'.
\tag{42}
$$

The integration in (41) is similar to the one in (36). As a result, using (38),

$$
\begin{aligned}
p(\mathbf{x}|l) = (2\pi)^{-\frac{d}{2}} A^l & \left(\frac{2\pi}{\kappa_{\mathbf{x}}^l}\right)^{\frac{d}{2}} 2^{\frac{d\left(\nu^l + n_t^l + 1\right)}{2}} \Gamma_d\left(\frac{\nu^l + n_t^l + 1}{2}\right) \\
& \left|\mathbf{T}_{\mathbf{x}}^l\right|^{\frac{\nu^l + n_t^l + 1}{2}} {}_2F_1\left(\frac{\nu^l + n_s^l}{2}, \frac{\nu^l + n_t^l + 1}{2}; \frac{\nu^l}{2}; \mathbf{T}_s^l \mathbf{F}^l \mathbf{T}_{\mathbf{x}}^l \mathbf{F}^{l'}\right).
\end{aligned}
\tag{43}
$$

By replacing the value of A^l, we have the effective class-conditional density. We denote $O_{\mathrm{OBTL}}(\mathbf{x}|l) = p(\mathbf{x}|l)$, since it is the objective function for the OBTL classifier. As such,

$$
\begin{aligned}
O_{\mathrm{OBTL}}(\mathbf{x}|l) = {} & \pi^{-\frac{d}{2}}\left(\frac{\kappa_{t,n}^l}{\kappa_{\mathbf{x}}^l}\right)^{\frac{d}{2}} \Gamma_d\left(\frac{\nu^l + n_t^l + 1}{2}\right) \\
& \times \Gamma_d^{-1}\left(\frac{\nu^l + n_t^l}{2}\right) \left|\mathbf{T}_{\mathbf{x}}^l\right|^{\frac{\nu^l + n_t^l + 1}{2}} \left|\mathbf{T}_t^l\right|^{-\frac{\nu^l + n_t^l}{2}} \\
& \times {}_2F_1\left(\frac{\nu^l + n_s^l}{2}, \frac{\nu^l + n_t^l + 1}{2}; \frac{\nu^l}{2}; \mathbf{T}_s^l \mathbf{F}^l \mathbf{T}_{\mathbf{x}}^l \mathbf{F}^{l'}\right) \\
& \times {}_2F_1^{-1}\left(\frac{\nu^l + n_s^l}{2}, \frac{\nu^l + n_t^l}{2}; \frac{\nu^l}{2}; \mathbf{T}_s^l \mathbf{F}^l \mathbf{T}_t^l \mathbf{F}^{l'}\right).
\end{aligned}
\tag{44}
$$

V. OPTIMAL BAYESIAN TRANSFER LEARNING CLASSIFIER

Let c_t^l be the prior probability that the target sample $\mathbf{x}$ belongs to the class $l \in \{1, \cdots, L\}$. Since $0 < c_t^l < 1$ and $\sum_{l=1}^{L} c_t^l = 1$, a Dirichlet prior is assumed:

$$
(c_t^1, \cdots, c_t^L) \sim \mathrm{Dir}(L, \xi_t),
\tag{45}
$$

where $\xi_t = (\xi_t^1, \cdots, \xi_t^L)$ are the concentration parameters, and $\xi_t^l > 0$ for $l \in \{1, \cdots, L\}$. As the Dirichlet distribution is a conjugate prior for the categorical distribution, upon observing $\mathbf{n} = (n_t^1, \cdots, n_t^L)$ data for class l in the target domain, the posterior has a Dirichlet distribution:

$$
\begin{aligned}
\pi^{\star} = (c_t^1, \cdots, c_t^L | \mathbf{n}) & \sim \mathrm{Dir}(L, \xi_t + \mathbf{n}) \\
& = \mathrm{Dir}(L, \xi_t^1 + n_t^1, \cdots, \xi_t^L + n_t^L),
\end{aligned}
\tag{46}
$$

with the posterior mean of c_t^l as

$$
\mathrm{E}_{\pi^{\star}}(c_t^l) = \frac{\xi_t^l + n_t^l}{N_t + \xi_t^0},
\tag{47}
$$

where $N_t = \sum_{l=1}^{L} n_t^l$ and $\xi_t^0 = \sum_{l=1}^{L} \xi_t^l$. As such, the optimal Bayesian transfer learning (OBTL) classifier for any new unlabeled sample $\mathbf{x}$ in the target domain is defined as

$$
\Psi_{\mathrm{OBTL}}(\mathbf{x}) = \arg\max_{l \in \{1, \cdots, L\}} \mathrm{E}_{\pi^{\star}}(c_t^l) O_{\mathrm{OBTL}}(\mathbf{x}|l),
\tag{48}
$$

which minimizes the expected error of the classifier. If we do not have any prior knowledge for the selection of classes, we use the same concentration parameter for all the classes: $\xi_t = (\xi, \cdots, \xi)$. Hence, if the number of samples in each class is the same, $n_t^1 = \cdots = n_t^L$, the first term $\mathrm{E}_{\pi^{\star}}$ is the same for all the classes and (48) is reduced to:

$$
\Psi_{\mathrm{OBTL}}(\mathbf{x}) = \arg\max_{l \in \{1, \cdots, L\}} O_{\mathrm{OBTL}}(\mathbf{x}|l).
\tag{49}
$$

VI. OBC IN TARGET DOMAIN

To see how the source data can help improve the performance, we compare the OBTL classifier with the OBC when there is only the target domain. Using exactly the same modeling and parameters as the previous sections, the priors for μ_t^l and $\mathbf{\Lambda}_t^l$, from (3) and (17), are given by

$$
\begin{aligned}
\mu_t^l | \mathbf{\Lambda}_t^l & \sim \mathcal{N}\left(\mathbf{m}_t^l, \left(\kappa_t^l \mathbf{\Lambda}_t^l\right)^{-1}\right), \\
\mathbf{\Lambda}_t^l & \sim W_d(\mathbf{M}_t^l, \nu^l).
\end{aligned}
\tag{50}
$$

Using Lemma 1, upon observing the dataset $\mathcal{D}_t^l$, the posteriors of μ_t^l and $\mathbf{\Lambda}_t^l$ will be

$$
\begin{aligned}
\mu_t^l | \mathbf{\Lambda}_t^l, \mathcal{D}_t^l & \sim \mathcal{N}\left(\mathbf{m}_{t,n}^l, \left(\kappa_{t,n}^l \mathbf{\Lambda}_t^l\right)^{-1}\right), \\
\mathbf{\Lambda}_t^l | \mathcal{D}_t^l & \sim W_d(\mathbf{M}_{t,n}^l, \nu_{t,n}^l),
\end{aligned}
\tag{51}
$$

where

$$
\kappa_{t,n}^l = \kappa_t^l + n_t^l, \quad \nu_{t,n}^l = \nu^l + n_t^l, \quad \mathbf{m}_{t,n}^l = \frac{\kappa_t^l \mathbf{m}_t^l + n_t^l \bar{\mathbf{x}}_t^l}{\kappa_t^l + n_t^l},
$$

$$
\left(\mathbf{M}_{t,n}^l\right)^{-1} = \left(\mathbf{M}_t^l\right)^{-1} + \mathbf{S}_t^l + \frac{\kappa_t^l n_t^l}{\kappa_t^l + n_t^l}(\mathbf{m}_t^l - \bar{\mathbf{x}}_t^l)(\mathbf{m}_t^l - \bar{\mathbf{x}}_t^l)',
\tag{52}
$$

with the corresponding sample mean and covariance:

$$
\bar{\mathbf{x}}_t^l = \frac{1}{n_t^l}\sum_{i=1}^{n_t^l} \mathbf{x}_{t,i}^l, \quad \mathbf{S}_t^l = \sum_{i=1}^{n_t^l}\left(\mathbf{x}_{t,i}^l - \bar{\mathbf{x}}_t^l\right)\left(\mathbf{x}_{t,i}^l - \bar{\mathbf{x}}_t^l\right)'.
\tag{53}
$$

Using (39) and similar integral steps, the effective class-conditional densities $p(\mathbf{x}|l) = O_{\text{OBC}}(\mathbf{x}|l)$ for the OBC are derived as [23]

$$O_{\text{OBC}}(\mathbf{x}|l) = \pi^{-\frac{d}{2}} \left(\frac{\kappa_{t,n}^l}{\kappa_{t,n}^l + 1} \right)^{\frac{d}{2}} \Gamma_d \left(\frac{\nu^l + n_t^l + 1}{2} \right)$$
$$\times \Gamma_d^{-1} \left(\frac{\nu^l + n_t^l}{2} \right) \left| \mathbf{M}_{\mathbf{x}}^l \right|^{\frac{\nu^l + n_t^l + 1}{2}} \left| \mathbf{M}_{t,n}^l \right|^{-\frac{\nu^l + n_t^l}{2}}, \quad (54)$$

where

$$\left(\mathbf{M}_{\mathbf{x}}^l \right)^{-1} = \left(\mathbf{M}_{t,n}^l \right)^{-1} + \frac{\kappa_{t,n}^l}{\kappa_{t,n}^l + 1} (\mathbf{m}_{t,n}^l - \mathbf{x})(\mathbf{m}_{t,n}^l - \mathbf{x})'. \quad (55)$$

The multi-class OBC [31], under a zero-one loss function, can be defined as

$$\Psi_{\text{OBC}}(\mathbf{x}) = \arg \max_{l \in \{1, \cdots, L\}} \mathrm{E}_{\pi^\star}(c_t^l) O_{\text{OBC}}(\mathbf{x}|l). \quad (56)$$

Similar to the OBTL, in the case of equal prior probabilities for the classes,

$$\Psi_{\text{OBC}}(\mathbf{x}) = \arg \max_{l \in \{1, \cdots, L\}} O_{\text{OBC}}(\mathbf{x}|l). \quad (57)$$

For binary classification, the definition of the OBC in (56) is equivalent to the definition in [23], where it is defined to be the binary classifier possessing the minimum Bayesian mean square error estimate [32] relative to the posterior distribution.

Theorem 6. *If* $\mathbf{M}_{ts}^l = \mathbf{0}$ *for all* $l \in \{1, \cdots, L\}$, *then*

$$\Psi_{\text{OBTL}}(\mathbf{x}) = \Psi_{\text{OBC}}(\mathbf{x}), \quad (58)$$

meaning that if there is no interaction between the source and target domains in all the classes a priori, then the OBTL classifier turns to the OBC classifier in the target domain.

Proof. If $\mathbf{M}_{ts}^l = \mathbf{0}$ for all $l \in \{1, \cdots, L\}$, then $\mathbf{F}^l = \mathbf{0}$. Since $_2F_1(a, b; c; \mathbf{0}) = 1$ for any values of a, b, and c, the Gauss hypergeometric functions will disappear in (44). From (31) and (52), $\mathbf{T}_t^l = \mathbf{M}_{t,n}^l$. From (42) and (55), $\mathbf{T}_{\mathbf{x}}^l = \mathbf{M}_{\mathbf{x}}^l$. As a result, $O_{\text{OBTL}}(\mathbf{x}|l) = O_{\text{OBC}}(\mathbf{x}|l)$, and consequently, $\Psi_{\text{OBTL}}(\mathbf{x}) = \Psi_{\text{OBC}}(\mathbf{x})$. $\square$

VII. LAPLACE APPROXIMATION OF GAUSS HYPERGEOMETRIC FUNCTION

We have derived the effective class-conditional densities in closed forms (44). However, deriving the OBTL classifier (48) requires computing the Gauss hypergeometric function of matrix argument. Computing the exact values of hypergeometirc functions of matrix argument using the series of zonal polynomial, as in (11), is time-consuming and is not scalable to high dimension. To facilitate computation, we propose to use the Laplace approximation of this function, as in [33], which is computationally efficient and scalable.

The Gauss hypergeomeric function has the following integral representation:

$$_2F_1(a, b; c; \mathbf{X}) = B_d^{-1}(a, c - a)$$
$$\times \int_{0_d < \mathbf{Y} < \mathbf{I}_d} |\mathbf{Y}|^{a - \frac{d+1}{2}} |\mathbf{I}_d - \mathbf{Y}|^{c-a-\frac{d+1}{2}} |\mathbf{I}_d - \mathbf{XY}|^{-b} d\mathbf{Y}, \quad (59)$$

which is valid under the following conditions: $\mathbf{X} \in \mathbf{C}^{d \times d}$ is symmetric and satisfies $\mathrm{Re}(\mathbf{X}) < \mathbf{I}_d$, $\mathrm{Re}(a) > \frac{d-1}{2}$, and $\mathrm{Re}(c - a) > \frac{d-1}{2}$. $B_d(\alpha, \beta)$ is the multivariate beta function

$$B_d(\alpha, \beta) = \frac{\Gamma_d(\alpha)\Gamma_d(\beta)}{\Gamma_d(\alpha + \beta)}, \quad (60)$$

where $\Gamma_d(\alpha)$ is the multivariate gamma function defined in (7). The Laplace approximation is one common solution to approximate the integral

$$I = \int_{y \in D} h(y) \exp(-\lambda g(y)) dy, \quad (61)$$

where $D \subseteq \mathbf{R}^d$ is an open set and λ is a real parameter. If $g(\lambda)$ has a unique minimum over D at point $\hat{y} \in D$, then the Laplace approximation to I is given by

$$\tilde{I} = (2\pi)^{\frac{d}{2}} \lambda^{-\frac{d}{2}} |g''(\hat{y})|^{-\frac{1}{2}} h(\hat{y}) \exp(-\lambda g(\hat{y})), \quad (62)$$

where $g''(y) = \frac{\partial^2 g(y)}{\partial y \partial y^T}$ is the Hessian of $g(y)$. The hypergeometric function $_2F_1(a, b; c; \mathbf{X})$ depends only on the eigenvalues of the symmetric matrix $\mathbf{X}$. Hence, without loss of generality, it is assumed that $\mathbf{X} = \mathrm{diag}\{x_1, \cdots, x_d\}$. The following g and h functions are used for (59):

$$g(\mathbf{Y}) = -a \log |\mathbf{Y}| - (c - a) \log |\mathbf{I}_d - \mathbf{Y}| + \log |\mathbf{I}_d - \mathbf{XY}|,$$
$$h(\mathbf{Y}) = B_d^{-1}(a, c - a) |\mathbf{Y}|^{-\frac{d+1}{2}} |\mathbf{I}_d - \mathbf{Y}|^{-\frac{d+1}{2}}. \quad (63)$$

Using (62) and (63), the Laplace approximation to $_2F_1(a, b; c; \mathbf{X})$ is given by [33]

$$_2\tilde{F}_1(a, b; c; \mathbf{X}) = \frac{2^{\frac{d}{2}} \pi^{\frac{d(d+1)}{4}}}{B_d(a, c - a)} J_{2,1}^{-\frac{1}{2}}$$
$$\times \prod_{i=1}^{d} \{\hat{y}_i^a (1 - \hat{y}_i)^{c-a} (1 - x_i \hat{y}_i)^{-b}\}, \quad (64)$$

where $\hat{y}_i$ is defined as

$$\hat{y}_i = \frac{2a}{\sqrt{\tau^2 - 4ax_i(c - b)} - \tau}, \quad (65)$$

with $\tau = x_i(b - a) - c$, and

$$J_{2,1} = \prod_{i=1}^{d} \prod_{j=i}^{d} \{a(1 - \hat{y}_i)(1 - \hat{y}_j) + (c - a)\hat{y}_i \hat{y}_j - b L_i L_j\}, \quad (66)$$

with

$$L_i = \frac{x_i \hat{y}_i (1 - \hat{y}_i)}{1 - x_i \hat{y}_i}. \quad (67)$$

The value of $_2F_1(a, b; c; \mathbf{X})$ at $\mathbf{X} = \mathbf{0}$ is 1, that is, $_2F_1(a, b; c; \mathbf{0}) = 1$. As a result, the Laplace approximation in (64) is calibrated at $\mathbf{X} = \mathbf{0}$ to give the calibrated Laplace approximation [33]:

$$_2\hat{F}_1(a, b; c; \mathbf{X}) = \frac{_2\tilde{F}_1(a, b; c; \mathbf{X})}{_2\tilde{F}_1(a, b; c; \mathbf{0})} = c^{cd - \frac{d(d+1)}{4}} R_{2,1}^{-\frac{1}{2}}$$
$$\times \prod_{i=1}^{d} \left\{ \left(\frac{\hat{y}_i}{a} \right)^a \left(\frac{1 - \hat{y}_i}{c - a} \right)^{c-a} (1 - x_i \hat{y}_i)^{-b} \right\}, \quad (68)$$

where

$$R_{2,1} = \prod_{i=1}^{d}\prod_{j=i}^{d}\left\{\frac{\hat{y}_i\hat{y}_j}{a} + \frac{(1-\hat{y}_i)(1-\hat{y}_j)}{c-a}\right.$$
$$\left. - \frac{bx_ix_j\hat{y}_i\hat{y}_j(1-\hat{y}_i)(1-\hat{y}_j)}{(1-x_i\hat{y}_i)(1-x_j\hat{y}_j)a(c-a)}\right\}. \tag{69}$$

VIII. EXPERIMENTS

A. Synthetic datasets

We have considered a simulation setup and evaluated the OBTL classifiers by the average classification error with different joint prior densities modeling the relatedness of the source and target domains. The setup is as follows. Unless mentioned, the feature dimension is $d = 10$, the number of classes in each domain is $L = 2$, the number of source training data per class is $n_s = n_s^l = 200$, the number of target training data per class is $n_t = n_t^l = 10$, $\nu = \nu^l = 25$, $\kappa_t = \kappa_t^l = 100$, $\kappa_s = \kappa_s^l = 100$, for both the classes $l = 1, 2$, $\mathbf{m}_t^1 = \mathbf{0}_d$, $\mathbf{m}_t^2 = 0.05 \times \mathbf{1}_d$, $\mathbf{m}_s^1 = \mathbf{m}_t^1 + \mathbf{1}_d$, and $\mathbf{m}_s^2 = \mathbf{m}_t^2 + \mathbf{1}_d$, where $\mathbf{0}_d$ and $\mathbf{1}_d$ are $d \times 1$ all-zero and all-one vectors, respectively. For the scale matrices, we choose $\mathbf{M}_t^l = k_t\mathbf{I}_d$, $\mathbf{M}_s^l = k_s\mathbf{I}_d$, and $\mathbf{M}_{ts}^l = k_{ts}\mathbf{I}_d$ for two classes $l = 1, 2$, where $\mathbf{I}_d$ is the $d \times d$ identity matrix. Note that choosing an identity matrix for $\mathbf{M}_{ts}^l$ makes sense when the order of the features in the two domains is the same. We have the constraint that the scale matrix $\mathbf{M}^l = \begin{pmatrix} \mathbf{M}_t^l, & \mathbf{M}_{ts}^l \\ \mathbf{M}_{ts}^l, & \mathbf{M}_s^l \end{pmatrix}$ should be positive definite for any class l. It is easy to check the following corresponding constraints on k_t, k_s, and k_{ts}: $k_t > 0$, $k_s > 0$, and $|k_{ts}| < \sqrt{k_t k_s}$. We define $k_{ts} = \alpha\sqrt{k_t k_s}$, where $|\alpha| < 1$. In this particular example, the value of $|\alpha|$ shows the amount of relatedness between the source and target domains. If $|\alpha| = 0$, the two domains are not related and if $|\alpha|$ is close to one, we have greater relatedness. We set $k_t = k_s = 1$ and plot the average classification error curves for different values of $|\alpha|$. All the simulations assume equal prior probabilities for the classes, so we use (49) and (57) for the OBTL classifier and OBC, respectively.

We evaluate the prediction performance according to the common evaluation procedure of Bayesian learning by average classification errors. To sample from the prior (2) we first sample from a Wishart distribution $W_{2d}(\mathbf{M}^l, \nu^l)$ to get a sample for $\mathbf{\Lambda}^l = \begin{pmatrix} \mathbf{\Lambda}_t^l, & \mathbf{\Lambda}_{ts}^l \\ \mathbf{\Lambda}_{ts}^l, & \mathbf{\Lambda}_s^l \end{pmatrix}$, for each class $l = 1, 2$, and then pick $(\mathbf{\Lambda}_t^l, \mathbf{\Lambda}_s^l)$, which is a joint sample from $p(\mathbf{\Lambda}_t^l, \mathbf{\Lambda}_s^l)$ in (16). Then given $\mathbf{\Lambda}_t^l$ and $\mathbf{\Lambda}_s^l$, we sample from (3) to get samples of μ_t^l and μ_s^l for $l = 1, 2$. Once we have μ_t^l, μ_s^l, $\mathbf{\Lambda}_t^l$, and $\mathbf{\Lambda}_s^l$, we generate 100 different training and test sets from (1). Training sets contain samples from both the target and source domains, but the test set contains only samples from the target domain. As the numbers of source and target training data per class is n_s and n_t, there are Ln_s and Ln_t source and target training data in total, respectively. We assume the size of the test set per class is 500 in the simulations, so 1000 in total. For each training and test set, we use the OBTL and its target-only version, OBC, and calculate the error. Then we average all the errors for 100 different training and test sets. We further repeat this whole process 100 times for different realizations of $\mathbf{\Lambda}_t^l$ and $\mathbf{\Lambda}_s^l$, μ_t^l, and μ_s^l for $l = 1, 2$, and finally average all
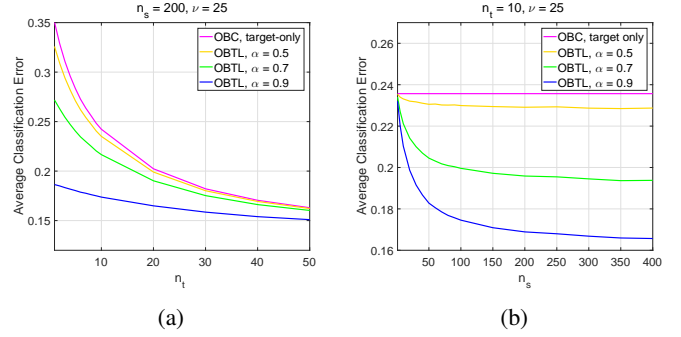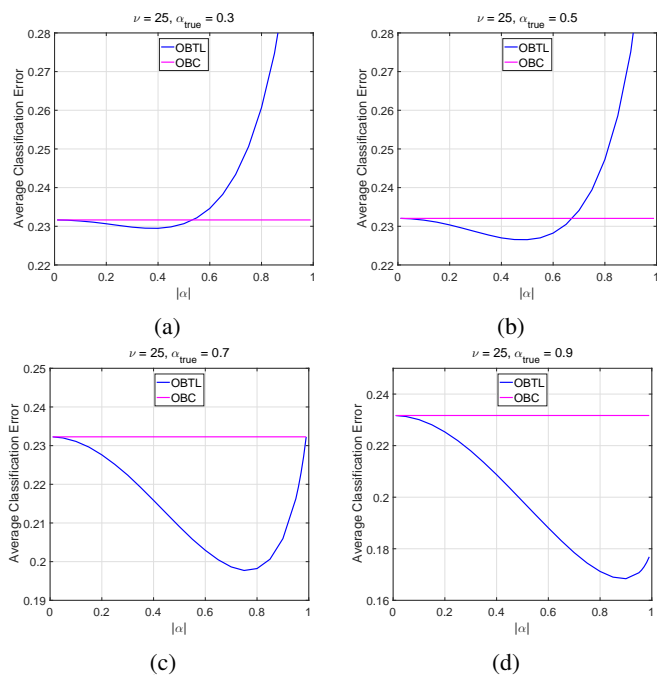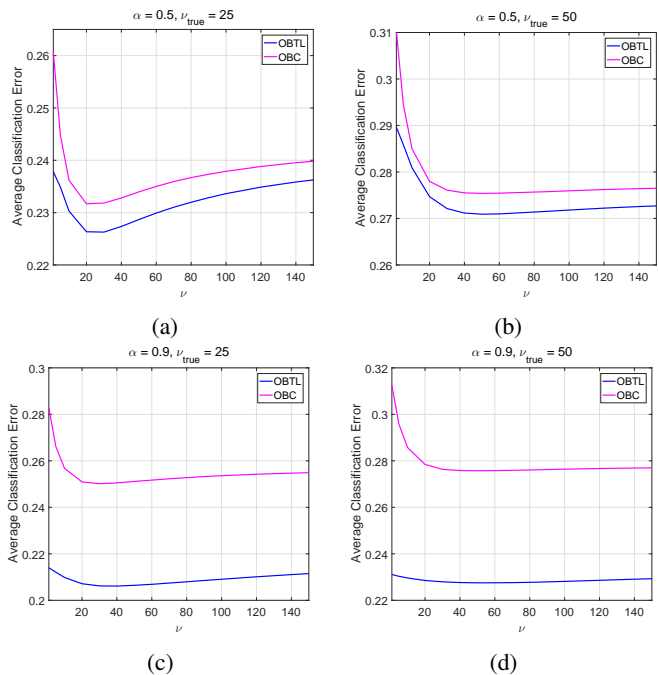

Figure 2: (a) Average classification error versus the number of target training data per class, n_t, (b) Average classification error versus the number of source training data per class, n_s.

the errors and return the average classification error. Note that in all figures, the hyperparameters used in the OBTL classifier are the same as the ones used for simulating data, except for the figures showing the sensitivity of the performance with respect to different hyperparameters, in which case we assume that true values of the hyperparameters used for simulating data are unknown.

To examine how the source data improves the classifier in target domain, we compare the performance of the OBTL classifier with the OBC designed in the target domain alone. The average classification error versus n_t is depicted in Fig. 2a for the OBC and OBTL with different values of α. When α is close to one, the performance of the OBTL classifier is much better than that of the OBC, this due to the greater relatedness between the two domains and appropriate use of the source data. This performance improvement is especially noticeable when n_t is small, which reflects the real-world scenario. In Fig. 2a, we also observe that the errors of the OBTL classifier and OBC are converging to a similar value when n_t gets very large, meaning that the source data are redundant when there is a large amount of target data. When α is larger, the error curves converge faster to the optimal error, which is the average Bayes error of the target classifier. Recall that when $\alpha = 0$, the OBTL classifier reduces to the OBC. In this particular example, the sign of α does not matter in the performance of the OBTL, which can be verified by (44). Hence, we can use $|\alpha|$ in all the cases.

Figure 2b depicts average classification error versus n_s for the OBC and OBTL with different values of α. The error of the OBC is constant for all n_s as it does not employ the source data. The error of the OBTL classifier equals that of the OBC when $n_s = 0$ and starts to decrease as n_s increases. In Fig. 2b when α is larger, the rate of improvement is greater since the two domains are more related.

We investigate the sensitivity of the OBTL with respect to the hyperparameters. Fig. 3 represents the average classification error of the OBTL with respect to $|\alpha|$, where we assume that we do not know the true value α_{true} of the amount of relatedness between source and target domains. In Figs. 3a-3d we plot the error curves when $\alpha_{true} = 0.3, 0.5, 0.7, 0.9$, respectively. We observe several important trends in these figures. First of all, the performance gain of the OBTL

Figure 3: Average classification error vs $|\alpha|$

Figure 4: Average classification error vs ν

Figure 5: Average classification error vs κ_t

towards the OBC depends heavily on the relatedness (value of α_{true}) of source and target and the value of α used in the classifier. Generally speaking, there exists an α_{max} in $(0, 1)$ such that for $|\alpha| < \alpha_{max}$, the OBTL has a performance gain towards the OBC, where maximum gain is achieved at $|\alpha| = \alpha_{true}$. Second, the performance gain is higher when the two domains are highly related (Fig. 3d). Third, when the two domains are very related, for example, $\alpha_{true} = 0.9$ in Fig. 3d, $\alpha_{max} = 1$, meaning that for any $|\alpha|$, the OBTL has performance gain towards the target-only OBC. However, when the source and target domains are not related much, like Figs. 3a-3b, $\alpha_{max} < 1$, and choosing $|\alpha|$ greater than α_{max} leads to performance loss compared to the OBC. This means that exaggeration in the amount of relatedness between the two domains can hurt the transfer learning classifier when the two domains are not actually related, which refers to the concept of negative transfer.

Figure 4 shows the errors versus ν, assuming unknown true value ν_{true}, for different values of α (0.5 and 0.9) and ν_{true} (25 and 50). The salient point here is that the performance of the OBTL classifier is not so sensitive to ν if it is chosen in its allowable range, that is, $\nu \geq 2d$. In Fig. 4, the error of the OBTL does not change much for $\nu \geq 2d = 20$. As a result, we can choose any arbitrary $\nu \geq 2d$ in real datasets without worrying about critical performance deterioration.

Figure 5 depicts average classification error versus κ_t for two different values of α (0.5 and 0.9), where the true value of κ_t is $\kappa_{true} = 50$. Similar to ν, if κ_t is greater than a value (20 in Fig. 5), the performance does not change much. According to (31), it is better to choose κ_t^l and κ_s^l to be proportional to n_t and n_s, respectively, since the values of updated means $\mathbf{m}_{t,n}^l$ and $\mathbf{m}_{s,n}^l$ are weighted averages of our prior knowledge about means, $\mathbf{m}_t^l$ and $\mathbf{m}_s^l$, and the sample means $\bar{\mathbf{x}}_t^l$ and $\bar{\mathbf{x}}_s^l$. Assuming that $\kappa_t = \beta_t n_t$ and $\kappa_s = \beta_s n_s$, for some $\beta_t, \beta_s > 0$,
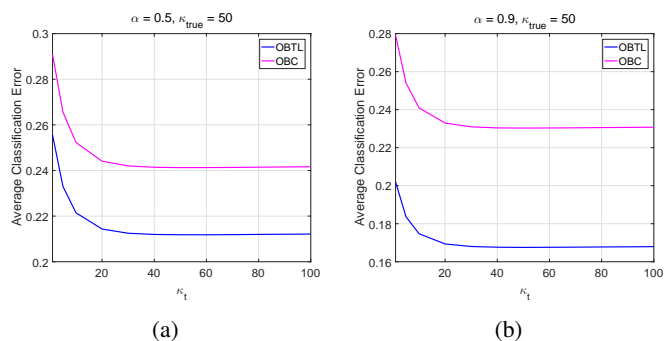
if we have higher trust on our priors on means, we pick higher β_t and β_s (like in Fig. 5); but for the untrustworthy priors, we choose lower values for β_t and β_s.

Sensitivity results in Figs. 3, 4, and 5 reveal that in our simulation setup the performance improvement of the OBTL depends on the value of α and true relatedness (α_{true} in this example) between the two domains and is not affected that much by the choices of other hyperparameters like ν, κ_t, and κ_s. We could have a reasonable range of α to get improved performance but the correct estimates of *relatedness* or *transferability* are critical, which is an important future research direction (see Conclusions in Section IX).

B. Real-world benchmark datasets

We test the OBTL classifier on *Office* [34] and *Caltech256* [35] image datasets, which have been adopted to help benchmark different transfer learning algorithms in the literature. We have used exactly the same evaluation setup and data splits of MMDT (Max-Margin Domain Transform) [10].

• **Office dataset:** This dataset has images in three different domains: *amazon*, *webcam*, and *dslr*. The dataset contains

Table I: Semi-supervised accuracy for different source and target domains in the *Office+Clatech256* dataset using SURF features. Domain names are denoted as a: *amazon*, w: *webcam*, d: *dslr*, c: *Caltech256*. The numbers in red show the best accuracy and the numbers in blue show the second best accuracy in each column. The results of the first six methods have been adopted from [17]. Similar to [17], we have also used the evaluation setup of [10] for the OBTL.

	a → w	a → d	a → c	w → a	w → d	w → c	d → a	d → w	d → c	c → a	c → w	c → d	Mean
1-NN-t	34.5	33.6	19.7	29.5	35.9	18.9	27.1	33.4	18.6	29.2	33.5	34.1	29.0
SVM-t	63.7	57.2	32.2	46.0	56.5	29.7	45.3	62.1	32.0	45.1	60.2	56.3	48.9
HFA [9]	57.4	55.1	31.0	**56.5**	56.5	29.0	42.9	60.5	30.9	43.8	58.1	55.6	48.1
MMDT [10]	64.6	56.7	36.4	47.7	67.0	32.2	46.9	74.1	34.1	49.4	63.8	56.5	52.5
CDLS [12]	68.7	60.4	35.3	51.8	60.7	33.5	50.7	68.5	34.9	50.9	66.3	59.8	53.5
ILS (1-NN) [17]	59.7	49.8	43.6	54.3	70.8	38.6	55.0	80.1	41.0	55.1	62.9	56.2	55.6
OBTL	72.3	60.8	43.1	55.0	75.8	38.1	54.3	85.2	40.6	54.7	70.6	61.4	59.3

Table II: The values of hyperparameters of the OBTL used in each experiment. n_t and n_s are based on the data splits provided by [10].

	a → w	a → d	a → c	w → a	w → d	w → c	d → a	d → w	d → c	c → a	c → w	c → d
n_t	3	3	3	3	3	3	3	3	3	3	3	3
n_s	20	20	20	8	8	8	8	8	8	8	8	8
β_t	1	2	5	1	1	3	1	2	2	1	1	1
β_s	1	2	5	1	1	3	1	2	2	1	1	1
k_t	0.01	0.005	0.002	0.005	0.01	0.005	0.005	0.02	0.003	0.004	0.004	0.005
k_s	0.01	0.004	0.002	0.005	0.01	0.005	0.005	0.02	0.003	0.004	0.004	0.005
α	0.7	0.6	0.99	0.9	0.99	0.99	0.99	0.99	0.99	0.8	0.8	0.7

31 classes including the office stuff like backpack, chair, keyboard, etc. The three domains *amazon*, *webcam*, and *dslr* contain images from Amazon's website, a webcam, and a digital single-lens reflex (dslr) camera, respectively, with different lighting and backgrounds. SURF [36] image features are used in all the domains, which are of dimension 800.

• **Office + Caltech256 dataset:** This dataset has $L = 10$ common classes of both *Office* and *Caltech256* datasets with the same feature dimension $d = 800$. Accordingt to the data splits of [10], the numbers of training data per class in the source domain are $n_s = 20$ for *amazon* and $n_s = 8$ for the other three domains, and in the target domain $n_t = 3$ for all the four domains. For this four-domain dataset, 20 random train-test splits have been created by [10]. We run the OBTL classifier on that 20 provided train-test splits and report the average accuracy. Note that the test data are solely from the target domains. Authors of MMDT [10] reduce the dimension to $d = 20$ using PCA. We follow the same procedure for the OBTL classifier.

Following the comparison framework of [17], which used the same evaluation setup of [10], we compare the OBTL's performance in terms of accuracy (10-class) in Table 1 with two target-only classifiers and four state-of-the-art semi-supervised transfer learning algorithms (including [17] itself). The evaluation setup is exactly the same for the OBTL and all the other six methods. As a result, we use the results of [17] for the first six methods in Table 1 and compare them with the OBTL classifier. The six methods are as follows.

• **1-NN-t and SVM-t:** The Nearest Neighbor (1-NN) and linear SVM classifiers designed using only the target data.

• **HFA [9]:** This Heterogeneous Feature Augmentation (HFA) method learns a common latent space between source and target domains using the max-margin approach and designs a classifier in that common space.

• **MMDT [10]:** This Max-Margin Domain Transform (MMDT) method learns a transformation between the source

and target domains and employs the weighted SVM for classification.

• **CDLS [12]:** This Cross-Domain Landmark Selection (CDLS) is a semi-supervised heterogeneous domain adaptation method, which derives a domain-invariant feature space for improved classification performance.

• **ILS (1-NN) [17]:** This is a recent method that learns an Invariant Latent Space (ILS) to reduce the discrepancy between the source and target domains and uses Riemannian optimization techniques to match statistical properties between samples projected into the latent space from different domains.

In Table 1, we have calculated the accuracy of the OBTL classifier in 12 distinct experiments, where the source-target pairs are different (source → target) in each experiment. We have marked the best accuracy in each column with red and the second best accuracy with blue. In 6 out of 12 cases, the OBTL has the best accuracy. In the other 6 cases, the OBTL has the second best accuracy, but its accuracy is very close to the best accuracy. We have written the mean accuracy of each method in the last column, which has been averaged over all the 12 different experiments. The OBTL classifier has the best mean accuracy and the ILS [17] has the second best accuracy among all the methods.

• **Hyperparameters of the OBTL:** We assume the same values of hyperparameters for all the 10 classes in each domain, so we can drop the superscript l denoting the class label. We set $\nu = 10d = 200$ for all the experiments. We choose α separately in each experiment since the relatedness between distinct pairs of domains are different. For $\mathbf{m}_t$ and $\mathbf{m}_s$, we pool all the target and source data in all the 10 classes, respectively, and use the sample means of the datasets. We find reasonable values for k_t, k_s, α, β_t ($\kappa_t = \beta_t n_t$), and β_s ($\kappa_s = \beta_s n_s$) using a cross-validation method. We show the values of those hyperparameters used in each experiment in Table 2. We have assumed equal prior probabilities for all the classes and used (49) for the OBTL classifier.

IX. Conclusions And Future Works

We have constructed a Bayesian transfer learning framework to tackle the semi-supervised and supervised transfer learning problems. The proposed Optimal Bayesian Transfer Learning (OBTL) classifier can deal with the lack of labeled data in the target domain and is optimal in this new Bayesian framework since it minimizes the expected classification error. We have obtained the closed-form posterior distribution of the target parameters and accordingly the closed-form effective class-conditional densities in the target domain to define the OBTL classifier. As the OBTL's objective function consists of hyper-geometric functions of matrix argument, we use the Laplace approximations of those functions to derive a computationally efficient and scalable OBTL classifier, while preserving its superior performance. We have compared the performance of the OBTL with its target-only version, OBC, to see how transferring from source to target domain can help. We have tested the OBTL classifier with real-world benchmark image datasets and demonstrated its excellent performance compared to other state-of-the-art domain adaption methods.

This paper considers a Gaussian model, in which we can derive closed-form solutions, as the case with the OBC. Since many practical problems cannot be approximated by a Gaussian model, an important aspect of OBC development has been the utilization of MCMC methods [37], [38]. These can be naturally extended to the OBTL setting, albeit, with greater computational burden.

We have only considered two domains in this paper, assuming there is only one source domain. Having seen the good performance of the OBTL classifier in two domains, in future work, we are going to apply it to the multi-source transfer learning problems, where we can benefit from the knowledge of different related sources in order to further improve the target classifier.

As in the case of the OBC, a basic engineering aspect of the OBTL is prior construction. This has been studied under different conditions in the context of the OBC: using the data from unused features to infer a prior distribution [39], deriving the prior distribution from models of the data-generating technology [37], deriving the prior from the uncertainties in the mathematical model characterizing the relevant physical system [40], and applying constraints based on prior knowledge to map the prior knowledge into a prior distribution via optimization [41], [42], [43]. The latter methods are very general and have been placed into a formal mathematical structure in [43], where the prior results from an optimization involving the Kullback-Leibler (KL) divergence constrained by conditional probability statements characterizing physical knowledge, such as genetic pathways in genomic medicine. A key focus of our future work will be to extend this general framework to the OBTL, which will require a formulation that incorporates knowledge relating the source and target domains. It should be emphasized that with optimal Bayesian classification, as well as with optimal Bayesian filtering [44], [45], [46], the prior distribution is not on the operator to be designed (classifier or filter) but on the underlying scientific model (feature-label distribution, covariance matrix, or obser-vation model) for which the operator is optimized. It is for this reason that uncertainty in the scientific model can be mapped into a prior distribution based on physical laws.

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[2] H. Venkateswara, S. Chakraborty, and S. Panchanathan, "Deep-learning systems for domain adaptation in computer vision: Learning transferable feature representations," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 117–129, 2017.

[3] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa, "Visual domain adaptation: A survey of recent advances," *IEEE signal processing magazine*, vol. 32, no. 3, pp. 53–69, 2015.

[4] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, no. 1, p. 9, 2016.

[5] G. Csurka, "Domain adaptation for visual applications: A comprehensive survey," *arXiv preprint arXiv:1702.05374*, 2017.

[6] N. Zou, Y. Zhu, J. Zhu, M. Baydogan, W. Wang, and J. Li, "A transfer learning approach for predictive modeling of degenerate biological systems," *Technometrics*, vol. 57, no. 3, pp. 362–373, 2015.

[7] P. Ganchev, D. Malehorn, W. L. Bigbee, and V. Gopalakrishnan, "Transfer learning of classification rules for biomarker discovery and verification from molecular profiling studies," *Journal of biomedical informatics*, vol. 44, pp. S17–S23, 2011.

[8] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2066–2073.

[9] L. Duan, D. Xu, and I. Tsang, "Learning with augmented features for heterogeneous domain adaptation," *ICML*, 2012.

[10] J. Hoffman, E. Rodner, T. Darrell, J. Donahue, and K. Saenko, "Efficient learning of domain-invariant image representations," in *International Conference on Learning Representations (ICLR)*, 2013.

[11] J. Hoffman, E. Rodner, J. Donahue, B. Kulis, and K. Saenko, "Asymmetric and category invariant feature transformations for domain adaptation," *International journal of computer vision*, vol. 109, no. 1-2, pp. 28–41, 2014.

[12] Y.-H. Hubert Tsai, Y.-R. Yeh, and Y.-C. Frank Wang, "Learning cross-domain landmarks for heterogeneous domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5081–5090.

[13] K. M. Borgwardt, A. Gretton, M. J. Rasch, H.-P. Kriegel, B. Schölkopf, and A. J. Smola, "Integrating structured biological data by kernel maximum mean discrepancy," *Bioinformatics*, vol. 22, no. 14, pp. e49–e57, 2006.

[14] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 193–200.

[15] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank, "Domain transfer svm for video concept detection," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1375–1381.

[16] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A dasvm classification technique and a circular validation strategy," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 5, pp. 770–787, 2010.

[17] S. Herath, M. Harandi, and F. Porikli, "Learning an invariant hilbert space for domain adaptation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 3956–3965.

[18] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1853–1865, Sept 2017.

[19] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International Conference on Machine Learning*, 2015, pp. 97–105.

[20] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 136–144.

[21] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[22] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Advances in neural information processing systems*, 2016, pp. 469–477.

[23] L. A. Dalton and E. R. Dougherty, "Optimal classifiers with minimum expected error within a Bayesian framework—Part I: Discrete and gaussian models," *Pattern Recognition*, vol. 46, no. 5, pp. 1301 – 1314, 2013.

[24] ——, "Optimal classifiers with minimum expected error within a Bayesian framework — Part II: Properties and performance analysis," *Pattern Recognition*, vol. 46, no. 5, pp. 1288 – 1300, 2013.

[25] R. J. Muirhead, *Aspects of multivariate statistical theory*. John Wiley & Sons, 2009.

[26] K. Halvorsen, V. Ayala, and E. Fierro, "On the marginal distribution of the diagonal blocks in a blocked Wishart random matrix," *International Journal of Analysis*, vol. 2016, pp. 1–5, 2016.

[27] D. K. Nagar and J. C. Mosquera-Benıtez, "Properties of matrix variate hypergeometric function distribution," *Applied Mathematical Sciences*, vol. 11, no. 14, pp. 677–692, 2017.

[28] A. G. Constantine, "Some non-central distribution problems in multivariate analysis," *Ann. Math. Statist.*, vol. 34, no. 4, pp. 1270–1285, 12 1963.

[29] D. K. Nagar and S. Nadarajah, "Appell's hypergeometric functions of matrix arguments," *Integral Transforms and Special Functions*, vol. 28, no. 2, pp. 91–112, 2017.

[30] A. K. Gupta, D. K. Nagar, and L. E. Sánchez, "Properties of matrix variate confluent hypergeometric function distribution," *Journal of Probability and Statistics*, vol. 2016, 2016.

[31] L. A. Dalton and M. R. Yousefi, "On optimal Bayesian classification and risk estimation under multiple classes," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2015, no. 1, p. 8, 2015.

[32] L. A. Dalton and E. R. Dougherty, "Bayesian minimum mean-square error estimation for classification error-Part I: Definition and the Bayesian MMSE error estimator for discrete classification," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 115–129, Jan 2011.

[33] R. W. Butler and A. T. A. Wood, "Laplace approximations for hypergeometric functions with matrix argument," *The Annals of Statistics*, vol. 30, no. 4, pp. 1155–1177, 2002.

[34] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 213–226.

[35] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," *Technical Report 7694, California Institute of Technology*, 2007.

[36] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Computer vision–ECCV 2006*, pp. 404–417, 2006.

[37] J. M. Knight, I. Ivanov, and E. R. Dougherty, "MCMC implementation of the optimal Bayesian classifier for non-Gaussian models: Model-based RNA-seq classification," *BMC bioinformatics*, vol. 15, no. 1, p. 401, 2014.

[38] J. M. Knight, I. Ivanov, K. Triff, R. S. Chapkin, and E. R. Dougherty, "Detecting multivariate gene interactions in RNA-seq data using optimal Bayesian classification," *IEEE/ACM transactions on computational biology and bioinformatics*, 2015.

[39] L. A. Dalton and E. R. Dougherty, "Application of the Bayesian MMSE estimator for classification error to gene expression microarray data," *Bioinformatics*, vol. 27, no. 13, pp. 1822–1831, 2011.

[40] A. Zollanvari and E. R. Dougherty, "Incorporating prior knowledge induced from stochastic differential equations in the classification of stochastic observations," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2016, no. 1, p. 2, 2016.

[41] M. S. Esfahani and E. R. Dougherty, "Incorporation of biological pathway knowledge in the construction of priors for optimal Bayesian classification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 1, pp. 202–218, Jan 2014.

[42] ——, "An optimization-based framework for the transformation of incomplete biological knowledge into a probabilistic structure and its application to the utilization of gene/protein signaling pathways in discrete phenotype classification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 6, pp. 1304–1321, Nov 2015.

[43] S. Boluki, M. S. Esfahani, X. Qian, and E. R. Dougherty, "Incorporating biological prior knowledge for bayesian learning via maximal knowledge-driven information priors," *BMC Bioinformatics*, vol. 18, no. 14, p. 552, Dec 2017.

[44] L. A. Dalton and E. R. Dougherty, "Intrinsically optimal Bayesian robust filtering," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 657–670, Feb 2014.

[45] X. Qian and E. R. Dougherty, "Bayesian regression with network prior: Optimal Bayesian filtering perspective," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6243–6253, Dec 2016.

[46] R. Dehghannasiri, M. S. Esfahani, and E. R. Dougherty, "Intrinsically Bayesian robust Kalman filter: An innovation process approach," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2531–2546, May 2017.