Kernelized Bayesian Transfer Learning*

Mehmet Gönen[†]
mehmet.gonen@sagebase.org
Sage Bionetworks
Seattle, WA 98109, USA

Adam A. Margolin[†]
adam.margolin@sagebase.org
Sage Bionetworks
Seattle, WA 98109, USA

Abstract

Transfer learning considers related but distinct tasks defined on heterogenous domains and tries to transfer knowledge between these tasks to improve generalization performance. It is particularly useful when we do not have sufficient amount of labeled training data in some tasks, which may be very costly, laborious, or even infeasible to obtain. Instead, learning the tasks jointly enables us to effectively increase the amount of labeled training data. In this paper, we formulate a kernelized Bayesian transfer learning framework that is a principled combination of kernel-based dimensionality reduction models with task-specific projection matrices to find a shared subspace and a coupled classification model for all of the tasks in this subspace. Our two main contributions are: (i) two novel probabilistic models for binary and multiclass classification, and (ii) very efficient variational approximation procedures for these models. We illustrate the generalization performance of our algorithms on two different applications. In computer vision experiments, our method outperforms the state-of-the-art algorithms on nine out of 12 benchmark supervised domain adaptation experiments defined on two object recognition data sets. In cancer biology experiments, we use our algorithm to predict mutation status of important cancer genes from gene expression profiles using two distinct cancer populations, namely, patient-derived primary tumor data and in-vitro-derived cancer cell line data. We show that we can increase our generalization performance on primary tumors using cell lines as an auxiliary data source.

1 Introduction

In many real-life applications, obtaining sufficient amount of labeled training data to have a reliable predictor may be very costly, laborious, or even infeasible. Instead, we can make use of labeled training data available from related tasks to increase our generalization performance. Transfer learning (also known as domain adaptation or cross-domain learning) aims to transfer knowledge between related tasks defined on heterogenous domains (Pan and Yang 2010). Heterogeneity may be due to different feature representations or data distributions over the same set of features. This setup is significantly different from multitask learning where we are given tasks with data points from the same feature representation (Caruana 1997; Argyriou, Evgeniou, and Pontil 2008).

Transfer learning algorithms are well-suited for natural language processing, computer vision, and computational biology applications due to their inherent suitability for knowledge transfer. For example, text collections from different languages, image collections from different types of recording devices, or biospecimen collections from different tissue types are natural candidates for transfer learning.

1.1 Related Work

Blitzer, McDonald, and Pereira (2006) find correspondences among features from different tasks and learn a shared feature space using these correspondences. Daumé III (2007) replicates input features to produce shared and domain-specific features, which are jointly fed into a supervised method to perform domain adaptation implicitly. Jiang et al. (2008) formulate a *support vector machine* (SVM; Vapnik 1998) model that uses support vectors of the source domain to improve the generalization performance on the target domain. Duan et al. (2009; 2010) learn a cross-domain kernel function and an SVM model by jointly minimizing the structural risk of the classifier and the mismatch between data distributions of the two domains. Bergamo and Torresani (2010) exploit strongly-labeled target domain data to improve labeling of weakly-labeled source domain hence to improve knowledge transfer between the two domains using a transductive SVM model.

Dai et al. (2009) use a risk minimization framework that couples two Markov chains defined on labels and features of the source and target domains with different feature representations. Hoffman et al. (2013) learn a linear transformation to map target domain data points into the source domain and a multiclass classifier in the source domain jointly using a coupled optimization problem.

Gopalan, Li, and Chellappa (2011) propose a manifold learning approach that maps labeled data from source domain and unlabeled data from target domain on the Grassmann manifold to learn a classifier there. Gong et al. (2012)

[*] This study was financially supported by the Integrative Cancer Biology Program of the National Cancer Institute (grant no 1U54CA149237).

[†] Present address: Department of Biomedical Engineering, Oregon Health & Science University, Portland, OR 97239, USA.

formulate a geodesic flow kernel to directly exploit intrinsic structures of computer vision data sets when transferring knowledge between the domains.

Ben-David et al. (2007) learn a shared subspace by maximizing the margin on the labeled source domain data and minimizing the distance between the data distributions of the two domains. Argyriou, Maurer, and Pontil (2008) divide image classification tasks into groups and learn a shared subspace for each group to transfer knowledge. Saenko et al. (2010) learn a nonlinear transformation to find a shared subspace by mapping two data points from different domains as close as possible if they are from the same class and as distant as possible otherwise. Kulis, Saenko, and Darrell (2011) generalize the same idea to the asymmetric setting (i.e., different feature representations). Pan et al. (2011) find low-dimensional latent representations for data points from different domains in a shared subspace by minimizing the maximum mean discrepancy between domains and maximizing the dependence between labels and latent features.

Bahadori, Liu, and Zhang (2011) give a transductive large-margin optimization algorithm that projects data points from the source and target domains into a shared subspace by minimizing reconstruction and prediction losses jointly. Duan, Xu, and Tsang (2012) map data points from different domains into a shared subspace using separate projection matrices and augment the projected data points with original features before feeding them into a supervised learning algorithm such as an SVM. Han, Liao, and Carin (2012) formulate a probabilistic model that generates the original features of heterogeneous domains from their latent representations in a shared subspace and learns a joint probit classifier in this subspace.

1.2 Our Contribution

Previous methods have been proposed for transfer learning, but none of them offers a fully Bayesian solution to domain adaptation on heterogenous domains in a discriminative setting. In this paper, we choose to find a shared subspace between the tasks using task-specific kernel-based dimensionality reduction models (Schölkopf and Smola 2002; Shawe-Taylor and Cristianini 2004) and to learn a coupled linear classifier in this subspace by combining these two steps with a fully Bayesian framework. Our formulation shares some similarities:

(i) with Kulis, Saenko, and Darrell (2011) and Pan et al. (2011) due to the shared subspace between domains,

(ii) with Hoffman et al. (2013) due to the coupled classifier,

(iii) with Bahadori, Liu, and Zhang (2011), Duan, Xu, and Tsang (2012), and Han, Liao, and Carin (2012) due to both parts.

We discuss the differences between our method and these methods in Section 2.5 after giving a detailed description of our method, which can also be interpreted as the generalization of the *relevance vector machine* (RVM; Tipping 2001) to transfer learning setup.

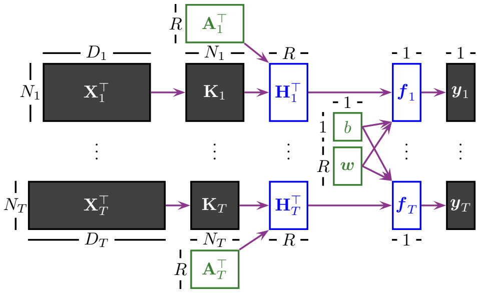

Figure 1: Flowchart of kernelized transfer learning for binary classification.

1.3 Preliminaries and Notation

We assume that there are T related binary classification tasks but their data points come from heterogeneous domains, namely, $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_T$. For each task, we are given an independent and identically distributed sample $\mathbf{X}_t = \{\boldsymbol{x}_{t,i} \in \mathcal{X}_t\}_{i \in \mathcal{I}_t}$ and a label vector $\boldsymbol{y}_t = \{y_{t,i} \in \{-1, +1\}\}_{i \in \mathcal{I}_t}$, where $\mathcal{I}_t$ gives the indices of data points in task t. There is a task-specific kernel function for each task to define similarities between the data points, i.e., $k_t \colon \mathcal{X}_t \times \mathcal{X}_t \to \mathbb{R}$, which is used to calculate the corresponding kernel matrix $\mathbf{K}_t = \{k_t(\boldsymbol{x}_{t,i}, \boldsymbol{x}_{t,j})\}_{i \in \mathcal{I}_t, j \in \mathcal{I}_t}$.

Figure 1 illustrates the method we propose to learn a conjoint model across the tasks; it is composed of two main parts: (i) projecting data points from different tasks into a shared subspace using a separate *kernel-based dimensionality reduction* model for each task and (ii) performing *coupled binary classification* in this subspace using a common set of classification parameters. We first briefly explain these two parts and introduce the notation used.

We first perform feature extraction using the input kernel matrices $\{\mathbf{K}_t \in \mathbb{R}^{N_t \times N_t}\}_{t=1}^T$ and the task-specific projection matrices $\{\mathbf{A}_t \in \mathbb{R}^{N_t \times R}\}_{t=1}^T$, where N_t is the number of data points in task t and R is the subspace dimensionality. After the projection, we obtain the hidden representations of data points in the shared subspace, i.e., $\{\mathbf{H}_t = \mathbf{A}_t^\top \mathbf{K}_t\}_{t=1}^T$. Using a kernel-based formulation has two main implications: (i) We can apply our method to tasks with very high dimensional representations. (ii) We can learn better subspaces using nonlinear or domain-specific kernel functions.

The coupled classification part calculates the predicted outputs $\{\boldsymbol{f}_t = \mathbf{H}_t^\top \boldsymbol{w} + \mathbf{1}b\}_{t=1}^T$ in the shared subspace using the same set of classification parameters $\{b \in \mathbb{R}, \boldsymbol{w} \in \mathbb{R}^R\}$. These outputs are mapped to labels by looking at their signs.

2 Kernelized Bayesian Transfer Learning

We formulate a probabilistic model, called *kernelized Bayesian transfer learning* (KBTL), for the method described earlier. We can derive a very efficient inference algorithm using variational approximation because our method combines the kernel-based dimensionality reduction and coupled binary classification parts with a fully conjugate probabilistic model.

Figure 2 gives the graphical model of KBTL with hyper-parameters, priors, latent variables, and model parameters. As described earlier, the main idea can be summarized as:

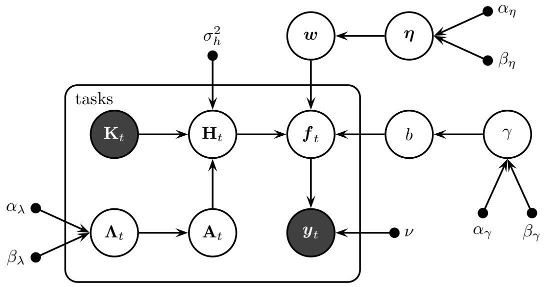

Figure 2: Graphical model of kernelized Bayesian transfer learning for binary classification.

(i) to find hidden representations for the data points of all tasks by mapping them into a shared subspace with the help of kernel matrices and task-specific projection matrices and (ii) to perform coupled binary classification in this subspace using a common set of classification parameters.

There are some additions to the notation described earlier: The $N_t \times R$ matrix of priors for the entries of the task-specific projection matrix $\mathbf{A}_t$ is denoted by $\boldsymbol{\Lambda}_t$. The $R \times 1$ vector of priors for the classification parameters $\boldsymbol{w}$ is denoted by $\boldsymbol{\eta}$. The prior for the bias parameter b is denoted by γ. For these three priors, there are three sets of hyper-parameters, namely, $\{\alpha_\lambda, \beta_\lambda\}$, $\{\alpha_\eta, \beta_\eta\}$, and $\{\alpha_\gamma, \beta_\gamma\}$. The standard deviation for the hidden representations is given as σ_h. As short-hand notations, the hyper-parameters are denoted by $\boldsymbol{\zeta} = \{\alpha_\eta, \beta_\eta, \alpha_\gamma, \beta_\gamma, \alpha_\lambda, \beta_\lambda, \sigma_h, \nu\}$, the priors by $\boldsymbol{\Xi} = \{\gamma, \boldsymbol{\eta}, \{\boldsymbol{\Lambda}_t\}_{t=1}^T\}$, and the latent variables and model parameters by $\boldsymbol{\Theta} = \{b, \boldsymbol{w}, \{\boldsymbol{f}_t, \mathbf{A}_t, \mathbf{H}_t\}_{t=1}^T\}$. Dependence on $\boldsymbol{\zeta}$ is omitted for clarity throughout the manuscript.

The distributional assumptions of the kernel-based dimensionality reduction part are defined as

$$\begin{aligned}
\lambda_{t,s}^i &\sim \mathcal{G}(\lambda_{t,s}^i; \alpha_\lambda, \beta_\lambda) && \forall (t,i,s) \\
a_{t,s}^i | \lambda_{t,s}^i &\sim \mathcal{N}(a_{t,s}^i; 0, (\lambda_{t,s}^i)^{-1}) && \forall (t,i,s) \\
h_{t,i}^s | \boldsymbol{a}_{t,s}, \boldsymbol{k}_{t,i} &\sim \mathcal{N}(h_{t,i}^s; \boldsymbol{a}_{t,s}^\top \boldsymbol{k}_{t,i}, \sigma_h^2) && \forall (t,s,i),
\end{aligned}$$

where the superscript indexes the rows and the subscript indexes the columns. $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents the normal distribution with the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$. $\mathcal{G}(\cdot; \alpha, \beta)$ denotes the gamma distribution with the shape parameter α and the scale parameter β.

The coupled binary classification part has the following distributional assumptions:

$$\begin{aligned}
\gamma &\sim \mathcal{G}(\gamma; \alpha_\gamma, \beta_\gamma) \\
b | \gamma &\sim \mathcal{N}(b; 0, \gamma^{-1}) \\
\eta_s &\sim \mathcal{G}(\eta_s; \alpha_\eta, \beta_\eta) && \forall s \\
w_s | \eta_s &\sim \mathcal{N}(w_s; 0, \eta_s^{-1}) && \forall s \\
f_{t,i} | b, \boldsymbol{w}, \boldsymbol{h}_{t,i} &\sim \mathcal{N}(f_{t,i}; \boldsymbol{w}^\top \boldsymbol{h}_{t,i} + b, 1) && \forall (t,i) \\
y_{t,i} | f_{t,i} &\sim \delta(f_{t,i} y_{t,i} > \nu) && \forall (t,i),
\end{aligned}$$

where the predicted outputs $\{\boldsymbol{f}_t\}_{t=1}^T$, similar to the discriminant outputs in SVMs, are introduced to make the inference procedures efficient (Albert and Chib 1993). The nonnegative margin parameter ν is introduced to resolve the scaling ambiguity and to place a low-density region between two classes, similar to the margin idea in SVMs, which is generally used for semi-supervised learning (Lawrence and Jordan 2005). $\delta(\cdot)$ represents the Kronecker delta function that returns 1 if its argument is true and 0 otherwise.

2.1 Inference Using Variational Bayes

To obtain an efficient inference mechanism, we formulate a deterministic variational approximation instead of a Gibbs sampling approach, which is computationally expensive (Gelfand and Smith 1990). The variational methods use a lower bound on the marginal likelihood using an ensemble of factored posteriors to find the joint parameter distribution (Beal 2003). We can write the factorable ensemble approximation of the required posterior as

$$\begin{aligned}
&p(\boldsymbol{\Theta}, \boldsymbol{\Xi} | \{\mathbf{K}_t, \boldsymbol{y}_t\}_{t=1}^T) \approx q(\boldsymbol{\Theta}, \boldsymbol{\Xi}) = \\
&\quad \prod_{t=1}^T \left[q(\boldsymbol{\Lambda}_t) q(\mathbf{A}_t) q(\mathbf{H}_t) \right] q(\gamma) q(\boldsymbol{\eta}) q(b, \boldsymbol{w}) \prod_{t=1}^T q(\boldsymbol{f}_t)
\end{aligned}$$

and define each factor in the ensemble just like its full conditional distribution:

$$\begin{aligned}
q(\boldsymbol{\Lambda}_t) &= \prod_{i \in \mathcal{I}_t} \prod_{s=1}^R \mathcal{G}(\lambda_{t,s}^i; \alpha(\lambda_{t,s}^i), \beta(\lambda_{t,s}^i)) \\
q(\mathbf{A}_t) &= \prod_{s=1}^R \mathcal{N}(\boldsymbol{a}_{t,s}; \mu(\boldsymbol{a}_{t,s}), \Sigma(\boldsymbol{a}_{t,s})) \\
q(\mathbf{H}_t) &= \prod_{i \in \mathcal{I}_t} \mathcal{N}(\boldsymbol{h}_{t,i}; \mu(\boldsymbol{h}_{t,i}), \Sigma(\boldsymbol{h}_{t,i})) \\
q(\gamma) &= \mathcal{G}(\gamma; \alpha(\gamma), \beta(\gamma)) \\
q(\boldsymbol{\eta}) &= \prod_{s=1}^R \mathcal{G}(\eta_s; \alpha(\eta_s), \beta(\eta_s)) \\
q(b, \boldsymbol{w}) &= \mathcal{N}\left(\begin{bmatrix} b \\ \boldsymbol{w} \end{bmatrix}; \mu(b, \boldsymbol{w}), \Sigma(b, \boldsymbol{w}) \right) \\
q(\boldsymbol{f}_t) &= \prod_{i \in \mathcal{I}_t} \mathcal{TN}(f_{t,i}; \mu(f_{t,i}), \Sigma(f_{t,i}), \rho(f_{t,i})),
\end{aligned}$$

where $\alpha(\cdot)$, $\beta(\cdot)$, $\mu(\cdot)$, and $\Sigma(\cdot)$ denote the shape parameter, the scale parameter, the mean vector, and the covariance matrix for their arguments, respectively. $\mathcal{TN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot))$ denotes the truncated normal distribution with the mean vector $\boldsymbol{\mu}$, the covariance matrix $\boldsymbol{\Sigma}$, and the truncation rule $\rho(\cdot)$ such that $\mathcal{TN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot)) \propto \mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ if $\rho(\cdot)$ is true and $\mathcal{TN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot)) = 0$ otherwise.

We can bound the marginal likelihood using Jensen's inequality:

$$\begin{aligned}
&\log p(\{\boldsymbol{y}_t\}_{t=1}^T | \{\mathbf{K}_t\}_{t=1}^T) \geq \\
&\quad \mathrm{E}_{q(\boldsymbol{\Theta}, \boldsymbol{\Xi})} \left[\log p(\{\boldsymbol{y}_t\}_{t=1}^T, \boldsymbol{\Theta}, \boldsymbol{\Xi} | \{\mathbf{K}_t\}_{t=1}^T) \right] \\
&\qquad - \mathrm{E}_{q(\boldsymbol{\Theta}, \boldsymbol{\Xi})} \left[\log q(\boldsymbol{\Theta}, \boldsymbol{\Xi}) \right]
\end{aligned}$$

and optimize this bound by maximizing with respect to each factor separately until convergence. The approximate posterior distribution of a specific factor $\boldsymbol{\tau}$ can be found as

$$q(\boldsymbol{\tau}) \propto \exp\Big(\mathrm{E}_{q(\{\boldsymbol{\Theta}, \boldsymbol{\Xi}\} \setminus \boldsymbol{\tau})} \left[\log p(\{\boldsymbol{y}_t\}_{t=1}^T, \boldsymbol{\Theta}, \boldsymbol{\Xi} | \{\mathbf{K}_t\}_{t=1}^T) \right] \Big).$$

For our proposed model, thanks to the conjugacy, the resulting approximate posterior distribution of each factor follows the same distribution as the corresponding factor.

2.2 Inference Details for Binary Classification

The approximate posterior distributions of the precision priors can be updated as

$$\begin{aligned}\alpha(\lambda_{t,s}^{i}) &= \alpha_\lambda + 1/2\\ \beta(\lambda_{t,s}^{i}) &= \big(1/\beta_\lambda + \langle (a_{t,s}^{i})^2\rangle/2\big)^{-1}\\ \alpha(\gamma) &= \alpha_\gamma + 1/2\\ \beta(\gamma) &= \big(1/\beta_\gamma + \langle b^2\rangle/2\big)^{-1}\\ \alpha(\eta_s) &= \alpha_\eta + 1/2\\ \beta(\eta_s) &= \big(1/\beta_\eta + \langle w_s^2\rangle/2\big)^{-1},\end{aligned}$$

where $\langle g(\cdot)\rangle$ denotes the posterior expectation as usual, i.e., $\mathrm{E}_{q(\cdot)}[g(\cdot)]$.

The approximate posterior distributions of the task-specific projection matrices can be updated as

$$\begin{aligned}\Sigma(\boldsymbol{a}_{t,s}) &= \big(\mathrm{diag}(\langle\boldsymbol{\lambda}_{t,s}\rangle) + \mathbf{K}_t\mathbf{K}_t^\top/\sigma_h^2\big)^{-1}\\ \mu(\boldsymbol{a}_{t,s}) &= \Sigma(\boldsymbol{a}_{t,s})\big(\mathbf{K}_t\langle(\boldsymbol{h}_t^s)^\top\rangle/\sigma_h^2\big)\end{aligned}$$

and the approximate posterior distribution of the hidden representation for each data point can be updated as

$$\begin{aligned}\Sigma(\boldsymbol{h}_{t,i}) &= \big(\mathbf{I}/\sigma_h^2 + \langle\boldsymbol{w}\boldsymbol{w}^\top\rangle\big)^{-1}\\ \mu(\boldsymbol{h}_{t,i}) &= \Sigma(\boldsymbol{h}_{t,i})\big(\langle\mathbf{A}_t^\top\rangle\boldsymbol{k}_{t,i}/\sigma_h^2 + \langle f_{t,i}\rangle\langle\boldsymbol{w}\rangle - \langle b\boldsymbol{w}\rangle\big).\end{aligned}$$

Note that the bias parameter b and the vector of weight parameters $\boldsymbol{w}$ are shared across the tasks.

The joint approximate posterior distribution of b and $\boldsymbol{w}$ can be updated as

$$\begin{aligned}\Sigma(b,\boldsymbol{w}) &= \begin{bmatrix}\langle\gamma\rangle + \sum\limits_{t=1}^{T} N_t & \sum\limits_{t=1}^{T}\mathbf{1}^\top\langle\mathbf{H}_t^\top\rangle\\ \sum\limits_{t=1}^{T}\langle\mathbf{H}_t\rangle\mathbf{1} & \mathrm{diag}(\langle\boldsymbol{\eta}\rangle) + \sum\limits_{t=1}^{T}\langle\mathbf{H}_t\mathbf{H}_t^\top\rangle\end{bmatrix}^{-1}\\ \mu(b,\boldsymbol{w}) &= \Sigma(b,\boldsymbol{w})\begin{bmatrix}\sum\limits_{t=1}^{T}\mathbf{1}^\top\langle\boldsymbol{f}_t\rangle\\ \sum\limits_{t=1}^{T}\langle\mathbf{H}_t\rangle\langle\boldsymbol{f}_t\rangle\end{bmatrix},\end{aligned}$$

where it can be seen that the inference mechanism transfers information between the tasks because they update $q(b,\boldsymbol{w})$ all together.

The approximate posterior distribution of the predicted outputs can be updated as

$$\begin{aligned}\Sigma(f_{t,i}) &= 1\\ \mu(f_{t,i}) &= \langle\boldsymbol{w}^\top\rangle\langle\boldsymbol{h}_{t,i}\rangle + \langle b\rangle\\ \rho(f_{t,i}) &\triangleq f_{t,i}y_{t,i} > \nu,\end{aligned}$$

where we can fortunately calculate the expectation of the truncated normal distribution in closed-form.

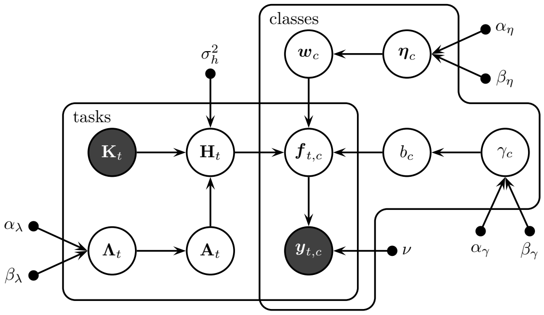

Figure 3: Graphical model of kernelized Bayesian transfer learning for multiclass classification.

2.3 Prediction

We can replace $p(\mathbf{A}_t|\{\mathbf{K}_u,\boldsymbol{y}_u\}_{u=1}^{T})$ with its approximate posterior distribution $q(\mathbf{A}_t)$ and obtain the predictive distribution of the hidden representation $\boldsymbol{h}_{t,\star}$ for a new data point $\boldsymbol{x}_{t,\star}$ as

$$\begin{aligned}&p(\boldsymbol{h}_{t,\star}|\boldsymbol{k}_{t,\star},\{\mathbf{K}_u,\boldsymbol{y}_u\}_{u=1}^{T}) =\\ &\prod_{s=1}^{R}\mathcal{N}(h_{t,\star}^{s};\mu(\boldsymbol{a}_{t,s})^\top\boldsymbol{k}_{t,\star},\sigma_h^2 + \boldsymbol{k}_{t,\star}^\top\Sigma(\boldsymbol{a}_{t,s})\boldsymbol{k}_{t,\star}).\end{aligned}$$

The predictive distribution of the predicted output $f_{t,\star}$ can also be found by replacing $p(b,\boldsymbol{w}|\{\mathbf{K}_t,\boldsymbol{y}_t\}_{t=1}^{T})$ with its approximate posterior distribution $q(b,\boldsymbol{w})$:

$$\begin{aligned}&p(f_{t,\star}|\boldsymbol{h}_{t,\star},\{\mathbf{K}_u,\boldsymbol{y}_u\}_{u=1}^{T}) =\\ &\mathcal{N}\left(f_{t,\star};\mu(b,\boldsymbol{w})^\top\begin{bmatrix}1\\ \boldsymbol{h}_{t,\star}\end{bmatrix}, 1 + [1\quad \boldsymbol{h}_{t,\star}]\Sigma(b,\boldsymbol{w})\begin{bmatrix}1\\ \boldsymbol{h}_{t,\star}\end{bmatrix}\right)\end{aligned}$$

and the predictive distribution of the class label $y_{t,\star}$ can be found using the predicted output distribution:

$$p(y_{t,\star} = +1|f_{t,\star},\{\mathbf{K}_u,\boldsymbol{y}_u\}_{u=1}^{T}) = \mathcal{Z}_{t,\star}^{-1}\Phi\left(\frac{\mu(f_{t,\star}) - \nu}{\Sigma(f_{t,\star})}\right),$$

where $\mathcal{Z}_{t,\star}$ is the normalization coefficient calculated for the test data point, and $\Phi(\cdot)$ is the standardized normal cumulative distribution function.

2.4 Multiclass Classification

In multiclass classification, we consider classification problems with more than two classes (i.e., $K > 2$). The most straightforward strategy is to train a distinct classifier for each class that separates this particular class from the remaining ones (i.e., one-versus-all classification). However, if we use our proposed method in such a setup, we would have different hidden representations for each class. Instead, we choose to learn a shared subspace for all classes and to apply one-versus-all classification strategy in this subspace.

Figure 3 gives the graphical model of KBTL for multiclass classification. There is a shared hidden representation space across the classes but each class has its own set of classification parameters $\{b_c \in \mathbb{R}, \boldsymbol{w}_c \in \mathbb{R}^R\}$ with their corresponding priors.

The distributional assumptions of the coupled classification part are modified as

$$\begin{aligned}
\gamma_c &\sim \mathcal{G}(\gamma_c; \alpha_\gamma, \beta_\gamma) && \forall c \\
b_c|\gamma_c &\sim \mathcal{N}(b_c; 0, \gamma_c^{-1}) && \forall c \\
\eta_{c,s} &\sim \mathcal{G}(\eta_{c,s}; \alpha_\eta, \beta_\eta) && \forall (c,s) \\
w_{c,s}|\eta_{c,s} &\sim \mathcal{N}(w_{c,s}; 0, \eta_{c,s}^{-1}) && \forall (c,s) \\
f_{t,c,i}|b_c, \boldsymbol{w}_c, \boldsymbol{h}_{t,i} &\sim \mathcal{N}(f_{t,c,i}; \boldsymbol{w}_c^\top \boldsymbol{h}_{t,i} + b_c, 1) && \forall (t,c,i) \\
y_{t,c,i}|f_{t,c,i} &\sim \delta(f_{t,c,i} y_{t,c,i} > \nu) && \forall (t,c,i).
\end{aligned}$$

By modifying the update equations of KBTL for binary classification, we can derive the update equations for the approximate posterior distributions of the classification parameters and predicted outputs, namely, $\{q(\gamma_c), q(\boldsymbol{\eta}_c), q(b_c, \boldsymbol{w}_c)\}_{c=1}^{K}$ and $\{q(\boldsymbol{f}_{t,c})\}_{t=1,c=1}^{T,K}$, as

$$\begin{aligned}
\alpha(\gamma_c) &= \alpha_\gamma + 1/2 \\
\beta(\gamma_c) &= \left(1/\beta_\gamma + \langle b_c^2 \rangle / 2\right)^{-1} \\
\alpha(\eta_{c,s}) &= \alpha_\eta + 1/2 \\
\beta(\eta_{c,s}) &= \left(1/\beta_\eta + \langle w_{c,s}^2 \rangle / 2\right)^{-1} \\
\Sigma(b_c, \boldsymbol{w}_c) &= \begin{bmatrix} \langle \gamma_c \rangle + \sum\limits_{t=1}^{T} N_t & \sum\limits_{t=1}^{T} \mathbf{1}^\top \langle \mathbf{H}_t^\top \rangle \\ \sum\limits_{t=1}^{T} \langle \mathbf{H}_t \rangle \mathbf{1} & \operatorname{diag}(\langle \boldsymbol{\eta}_c \rangle) + \sum\limits_{t=1}^{T} \langle \mathbf{H}_t \mathbf{H}_t^\top \rangle \end{bmatrix}^{-1} \\
\mu(b_c, \boldsymbol{w}_c) &= \Sigma(b_c, \boldsymbol{w}_c) \begin{bmatrix} \sum\limits_{t=1}^{T} \mathbf{1}^\top \langle \boldsymbol{f}_{t,c} \rangle \\ \sum\limits_{t=1}^{T} \langle \mathbf{H}_t \rangle \langle \boldsymbol{f}_{t,c} \rangle \end{bmatrix} \\
\Sigma(f_{t,c,i}) &= 1 \\
\mu(f_{t,c,i}) &= \left\langle \boldsymbol{w}_c^\top \right\rangle \langle \boldsymbol{h}_{t,i} \rangle + \langle b_c \rangle \\
\rho(f_{t,c,i}) &\triangleq f_{t,c,i} y_{t,c,i} > \nu.
\end{aligned}$$

The covariance update equation for the approximate posterior distribution of the hidden representation for each data point can be written as

$$\Sigma(\boldsymbol{h}_{t,i}) = \left(\mathbf{I}/\sigma_h^2 + \sum_{c=1}^{K} \left\langle \boldsymbol{w}_c \boldsymbol{w}_c^\top \right\rangle \right)^{-1}$$

and the mean update equation can be given as

$$\mu(\boldsymbol{h}_{t,i}) = \Sigma(\boldsymbol{h}_{t,i}) \left(\left\langle \mathbf{A}_t^\top \right\rangle \boldsymbol{k}_{t,i} / \sigma_h^2 + \sum_{c=1}^{K} \left(\langle f_{t,c,i} \rangle \langle \boldsymbol{w}_c \rangle - \langle b_c \boldsymbol{w}_c \rangle \right) \right),$$

where we use the classification parameters of all classes together. The update equations of the task-specific projection matrices and their priors remain intact.

2.5 Comparison to Related Work

Kulis, Saenko, and Darrell (2011) and Pan et al. (2011) first perform dimensionality reduction to find a shared subspace and then learn a classifier in this subspace. The dimensionality reduction step has its own target function different from the one that the classifier in the shared subspace uses. Hence, coupled training of these two steps as we do in our method may improve the overall system performance.

Hoffman et al. (2013) map data points from the source domain into the target domain, which requires learning a very large projection matrix for high-dimensional feature representations, which can be avoided by projecting both domains into low-dimensional spaces using, for example, principal component analysis (PCA) at the cost of information loss, whereas our method can work directly with high-dimensional feature representations using the kernel trick.

Bahadori, Liu, and Zhang (2011) and Duan, Xu, and Tsang (2012) combine dimensionality reduction and classification steps by formulating joint optimization problems, which are non-convex requiring time-consuming alternating optimization strategies and are limited to two domains, whereas our method can use more than two domains and is able to produce probabilistic outputs.

Han, Liao, and Carin (2012) formulate a generative model, which is limited to low-dimensional problems and is able to produce linear mappings due to its generative nature, and find a maximum a posteriori estimate of model parameters using an expectation-maximization algorithm. However, our discriminative model is able to scale to high-dimensional problems and to find nonlinear mappings using the kernel trick. Our method finds a full-Bayesian solution for its parameters using a variational approximation algorithm.

3 Experiments

We first test our new algorithm KBTL on 12 benchmark domain adaptation experiments derived from two computer vision data sets to illustrate its generalization performance and compare its results to previously reported results on these experiments. We then perform transfer learning experiments with heterogeneous populations for two different cancer types to show the suitability of our algorithm in a challenging and nonstandard application scenario. Our Matlab implementations for binary and multiclass classification are available at https://github.com/mehmetgonen/kbtl.

3.1 Computer Vision Experiments

In this first set of experiments, we use `Office` (Saenko et al. 2010) and `Caltech-256` (Griffin, Holub, and Perona 2007) data sets. `Office` data set contains images of office objects from 31 different categories (e.g., calculator, keyboard, monitor, mug, etc.) under three distinct domains: `amazon`, `webcam`, and `dslr`. The images from different domains have varying characteristics such as illumination and background. For example, `amazon` domain has "controlled" product images containing a single and centered object usually on a white background, whereas `webcam` and `dslr` domains have "uncontrolled" images with lighting variations and background changes. Heterogeneity of images across the domains can easily be seen from Figure 4 for `amazon`, `webcam`, and `dslr` domains. `Caltech-256` data set contains images from 256 categories under a diverse set of lighting conditions, poses, and backgrounds.

In our experiments, we use the ten common categories (i.e., backpack, bicycle, calculator, headphones, keyboard,

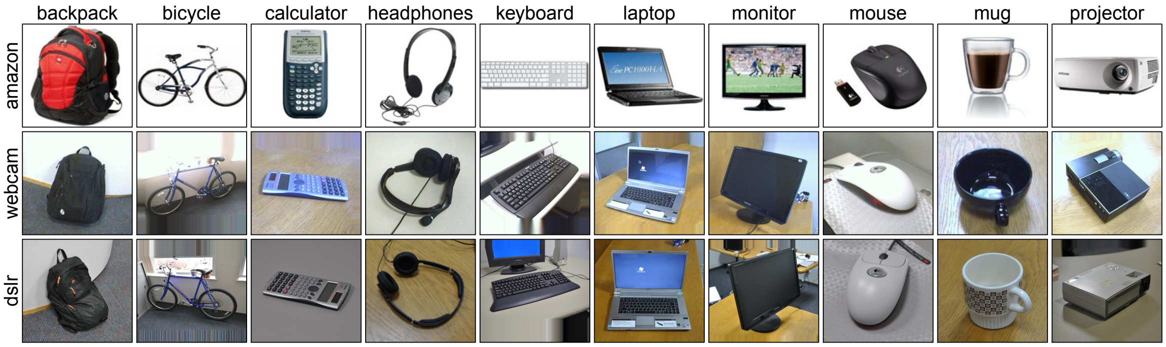

Figure 4: Sample images from `amazon`, `webcam`, and `dslr` domains in `Office` data set of Saenko et al. (2010).

Table 1: Multiclass classification accuracies for supervised domain adaptation experiments on the object recognition data sets. The results for six baseline algorithms are directly taken from Hoffman et al. (2013). The results of `KBTL(L)` and `KBTL(G)` are marked in bold face if they are better than all of the baseline algorithms.

Source	Target	SVMS	SVMT	ARCT	HFA	GFK	MMDT	KBTL(L)	KBTL(G)
webcam	amazon	35.7±0.4	45.6±0.7	43.4±0.5	45.9±0.7	44.1±0.4	47.7±0.9	**52.2±0.8**	**53.4±0.8**
dslr	amazon	34.0±0.3	45.7±0.9	42.5±0.5	45.8±0.9	45.7±0.6	46.9±1.0	**50.9±0.8**	**51.9±0.9**
caltech	amazon	35.9±0.4	45.3±0.9	44.1±0.6	45.5±0.9	44.7±0.8	49.4±0.8	**52.4±1.1**	**52.9±1.0**
amazon	webcam	33.9±0.7	62.4±0.9	55.7±0.9	61.8±1.1	58.6±1.0	64.6±1.2	**69.0±1.0**	**69.8±1.1**
dslr	webcam	74.3±0.5	62.1±0.8	78.3±0.5	62.1±0.7	76.5±0.5	74.1±0.8	69.1±0.9	70.0±1.0
caltech	webcam	30.8±1.1	60.3±1.0	55.9±1.0	60.5±0.9	63.7±0.8	63.8±1.1	**67.0±1.1**	**68.5±1.2**
amazon	dslr	35.0±0.8	55.9±0.8	50.2±0.7	52.7±0.9	50.7±0.8	56.7±1.3	**57.8±1.1**	**57.6±1.1**
webcam	dslr	66.6±0.7	55.1±0.8	71.3±0.8	51.7±1.0	70.5±0.7	67.0±1.1	60.8±1.0	61.8±1.3
caltech	dslr	35.6±0.7	55.8±0.9	50.6±0.8	51.9±1.1	57.7±1.1	56.5±0.9	57.4±1.3	**58.8±1.1**
amazon	caltech	35.1±0.3	32.0±0.8	37.0±0.4	31.1±0.6	36.0±0.5	36.4±0.8	35.8±0.8	35.9±0.7
webcam	caltech	31.3±0.4	30.4±0.7	31.9±0.5	29.4±0.6	31.1±0.6	32.2±0.8	**33.6±0.9**	**34.0±0.9**
dslr	caltech	31.4±0.3	31.7±0.6	33.5±0.4	31.0±0.5	32.9±0.5	34.1±0.8	**35.0±0.7**	**35.9±0.6**
Mean Performance		40.0±0.6	48.5±0.8	49.5±0.6	47.4±0.8	51.0±0.7	52.5±1.0	**53.4±0.9**	**54.2±1.0**

laptop, monitor, mouse, mug, projector) shared by `Office` and `Caltech-256`. We use 800-dimensional SURF-BoW features provided by Gong et al. (2012) for all four domains. To have results comparable to the previous studies, we follow the experimental setup used by Saenko et al. (2010), Gong et al. (2012), and Hoffman et al. (2013), and perform experiments for 20 random train/test splits provided by Hoffman et al. (2013). In 12 domain adaptation experiments defined between all possible pairs of the four domains, there are few labeled data points (three images) available for each category in the target domain, whereas we have much more labeled data points (20 images for `amazon` and eight images for others) in the source domain. The remaining data points from the target domain are used in the test phase. For each experiment, we report the mean and standard deviation of multiclass classification accuracies over 20 replications.

We compare eight algorithms (i.e., six baseline algorithms and our algorithm with two different kernels):

(i) `SVMS`: an SVM trained on the source domain,

(ii) `SVMT`: an SVM trained on the target domain,

(iii) `ARCT`: *asymmetric regularized cross-domain transformation* algorithm proposed by Kulis, Saenko, and Darrell (2011),

(iv) `HFA`: *heterogeneous feature adaptation* algorithm proposed by Duan, Xu, and Tsang (2012),

(v) `GFK`: *geodesic flow kernel algorithm* proposed by Gong et al. (2012),

(vi) `MMDT`: *max-margin domain transforms* algorithm proposed by Hoffman et al. (2013),

(vii) `KBTL(L)`: our algorithm for multiclass classification with the linear kernel,

(viii) `KBTL(G)`: our algorithm for multiclass classification with the Gaussian kernel.

The results for baseline algorithms are directly taken from Hoffman et al. (2013).

For our algorithm, the hyper-parameter values are selected as $(\alpha_\eta, \beta_\eta) = (\alpha_\gamma, \beta_\gamma) = (\alpha_\lambda, \beta_\lambda) = (1, 1), \sigma_h = 0.1$, and $\nu = 1$. The number of components in the hidden representation space is selected as $R = 20$. We take 200 iterations for variational inference scheme. For `KBTL(L)`, we calculate a linear kernel on each domain and normalize the kernel matrices to unit maximum value (i.e., dividing each kernel matrix by its maximum value) in order to eliminate scaling issues. For `KBTL(G)`, we calculate a Gaussian kernel on each domain, where we set the kernel width to the mean of pairwise distances between the training data points.

Table 2: Number of data points in source and target domains for transfer learning experiments.

		CCLE (source)		TCGA (target)	
Cancer	Gene	Mutant	WT	Mutant	WT
GBM	EGFR	5	36	47	102
	TP53	27	14	45	104
LUAD	TP53	34	9	79	90
	KRAS	17	26	46	123

Table 1 reports the multiclass classification accuracies for 12 supervised domain adaptation experiments. KBTL(L) outperforms all baseline algorithms on eight out of 12 experiments, whereas KBTL(G) outperforms them on nine experiments. They also improve the mean performance by 0.9 and 1.7 percentage units, respectively, compared to the baseline algorithm with the best performance. These results validate the better generalization performance of our algorithm irrespective of the kernel function used. Our algorithm does not perform well on two experiments defined between webcam and dslr. This can be explained by the similarity between these two domains as SVMS performs significantly better than SVMT on these experiments. That is why nearest-neighbor-based algorithms, i.e., ARCT and GFK, work better for such scenarios but not margin-based algorithms, i.e., HFA, MMDT, KBTL(L), and KBTL(G).

3.2 Cancer Biology Experiments

In this second set of experiments, we test if we can transfer information from in-vitro cancer cell line data to improve the accuracy of gene-expression-based predictors of the mutation status of important cancer genes in primary patient tumor samples. We use primary tumor data from The Cancer Genome Atlas (TCGA) (TCGA Research Network 2008) and cancer cell line data from the Cancer Cell Line Encyclopedia (CCLE) (Barretina et al. 2012). We define the learning task as predicting mutation status (Mutant versus Wild Type) of a particular gene using gene expression profiles. Specifically, we use expression profiles of 20,530 genes in TCGA from Illumina HiSeq experiments and expression profiles of 18,897 genes in CCLE from Affymetrix U133+2 microarrays. From TCGA, we use primary tumors from two cancer cohorts: glioblastoma multiforme (GBM) and lung adenocarcinoma (LUAD). From CCLE, we use cell lines with the corresponding tissue types: central nervous system glioma and lung adenocarcinoma. We perform four transfer learning experiments in which we try to transfer information from CCLE to TCGA (i.e., from cell lines to primary tumors). For both cancer types, we identify two genes that are most frequently mutated in TCGA among the ones listed in The Cancer Gene Census (Futreal et al. 2004). Table 2 summarizes the details of these four experiments.

We compare three algorithms:

(i) BRVM: *Bayesian relevance vector machine* algorithm proposed by Bishop and Tipping (2000) trained on the target domain (i.e., no transfer),

(ii) MMDT: *max-margin domain transforms* algorithm proposed by Hoffman et al. (2013) (i.e., the best baseline algorithm in the computer vision experiments),

(iii) KBTL: our algorithm for binary classification trained on both domains together.

For BRVM, we use our own Matlab implementation and set the hyper-parameter values to the default values as in KBTL, i.e., $(\alpha_\gamma, \beta_\gamma) = (\alpha_\lambda, \beta_\lambda) = (1, 1)$. For MMDT, we use the Matlab implementation provided by Hoffman et al. (2013) and reduce the dimensionality of each domain to 20 using PCA as suggested by Hoffman et al. (2013) to get reasonable run times for high-dimensional cancer data sets. For BRVM and KBTL, we calculate a linear kernel for each domain and normalize the kernel matrix to unit maximum value. For our algorithm, the number of components in the hidden representation space is selected as $R = 2$.

We perform experiments for 50 random train/test splits. For each replication, we randomly select 25 per cent of TCGA with stratification as the test set and use 25, 50, or 75 per cent as the training set, whereas we use all data points in CCLE for training. The training sets are normalized separately to have zero mean and unit standard deviation, and the test set in TCGA is then normalized using the mean and the standard deviation of the original training set in TCGA.

Figure 5 compares the performance of BRVM, MMDT, and KBTL in terms of the area under the ROC curve (AUC) values with varying training set size for the target domain on four transfer learning experiments using box-and-whisker plots. It also compares KBTL and the best baseline algorithm for each experiment using scatter plots. We clearly see that KBTL is statistically significantly superior to BRVM and MMDT in all of the experiments according to the paired t-test with $p < 0.05$. Note that knowledge transfer happens in all of the experiments for our algorithm (i.e., KBTL is consistently better than BRVM), whereas MMDT obtains worse AUC values than BRVM in some of the experiments (e.g., KRAS gene on LUAD cohort). These results show that KBTL is able to improve the predictive performance on primary tumors using cell lines as an auxiliary data source. Applications of this approach may have important clinical implications due to the difficulty and expense of obtaining data on primary tumors compared to cell lines.

4 Conclusions

We introduce a kernelized Bayesian transfer learning framework that can transfer information between tasks with heterogenous feature representations by mapping their data points into a shared subspace with task-specific projection matrices and learning a coupled classification model in this subspace. Our two main contributions are: (i) formulating novel probabilistic models that couple these two parts for binary and multiclass classification in a principled way and (ii) developing very efficient variational approximation procedures for these probabilistic models. We illustrate the practical importance of the method for two scenarios: (i) supervised domain adaptation experiments on object recognition data sets and (ii) transfer learning experiments with het-

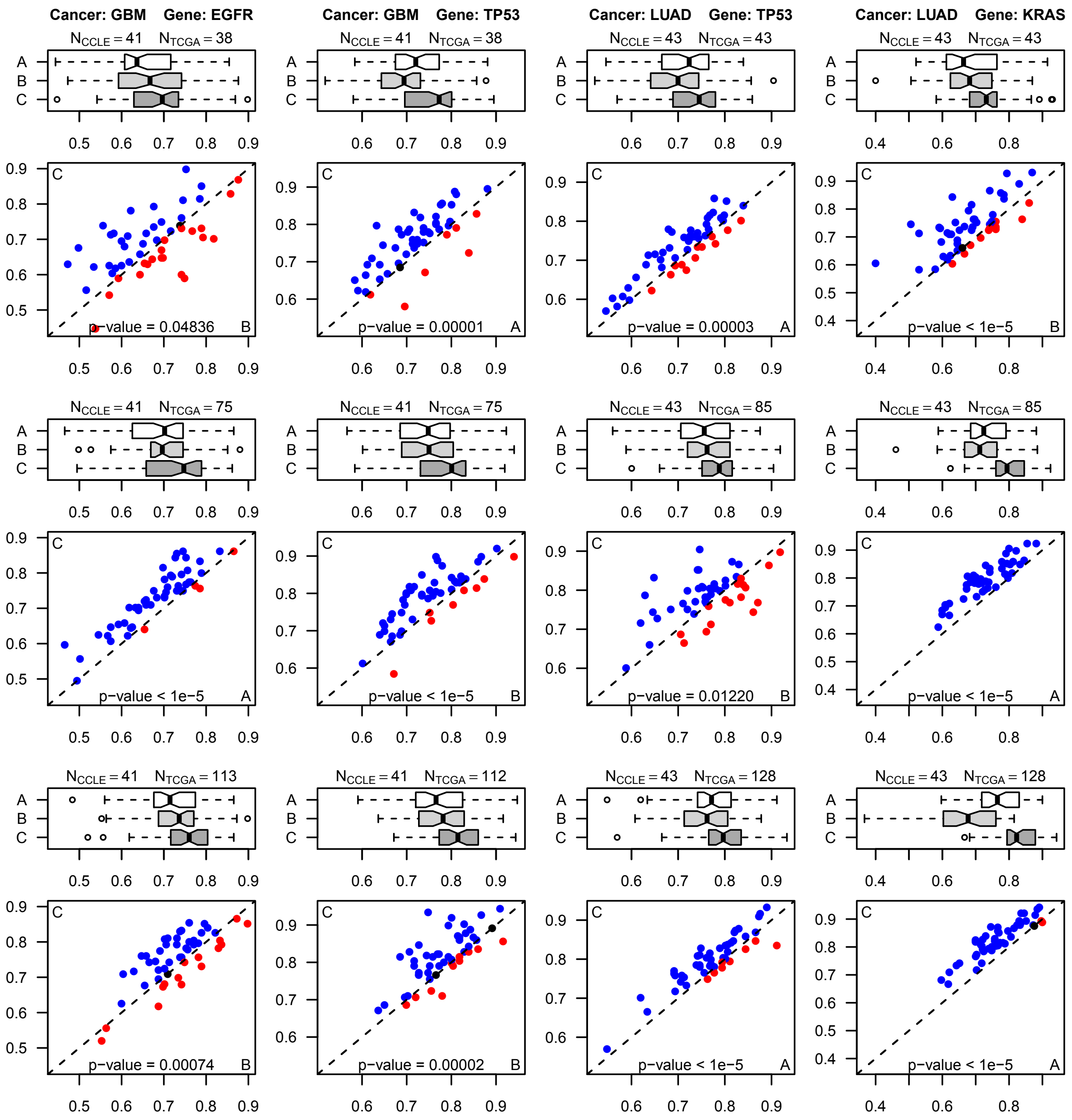

Figure 5: Comparison between (A) BRVM, (B) MMDT, and (C) KBTL in terms of AUC values on GBM and LUAD cohorts with varying training set size for the target domain.

erogenous populations for two cancer types. We show that our method outperforms the state-of-the-art domain adaptation algorithms on computer vision experiments. We also show that our method is able to transfer information between patient-derived primary tumor data and in-vitro-derived cancer cell line data.

Three interesting topics for future research are: (i) performing task grouping to cluster tasks with similar characteristics together, which leads to eliminating the negative transfer between tasks with different characteristics, (ii) extending coupled classification part of our algorithms using the low-density assumption of Lawrence and Jordan (2005) in order to make use of unlabeled data, which leads to semi-supervised variants of our algorithms, and (iii) combining multiple kernels to integrate different feature representations or similarities for each task to learn a better task-specific kernel function, known as *multiple kernel learning* (Gönen and Alpaydın 2011), using the formulation of Gönen (2012).

References

Albert, J. H., and Chib, S. 1993. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association* 88:669–679.

Argyriou, A.; Evgeniou, T.; and Pontil, M. 2008. Convex multi-task feature learning. *Machine Learning* 73(3):243–272.

Argyriou, A.; Maurer, A.; and Pontil, M. 2008. An algorithm for transfer learning in a heterogeneous environment. In *Pro-*

ceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, Part I, 71–85.

Bahadori, M. T.; Liu, Y.; and Zhang, D. 2011. Learning with minimum supervision: A general framework for transductive transfer learning. In *Proceedings of the 11th IEEE International Conference on Data Mining*, 61–70.

Barretina, J.; Caponigro, G.; Stransky, N.; Venkatesan, K.; Margolin, A. A.; et al. 2012. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* 483:603–607.

Beal, M. J. 2003. *Variational Algorithms for Approximate Bayesian Inference*. Ph.D. Dissertation, The Gatsby Computational Neuroscience Unit, University College London.

Ben-David, S.; Blitzer, J.; Crammer, K.; and Pereira, F. 2007. Analysis of representations for domain adaptation. In *Advances in Neural Information Processing Systems 19*, 137–144.

Bergamo, A., and Torresani, L. 2010. Exploiting weakly-labeled web images to improve object classification: A domain adaptation approach. In *Advances in Neural Information Processing Systems 23*, 181–189.

Bishop, C. M., and Tipping, M. E. 2000. Variational relevance vector machines. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, 46–53.

Blitzer, J.; McDonald, R.; and Pereira, F. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 120–128.

Caruana, R. 1997. Multitask learning. *Machine Learning* 28(1):41–75.

Dai, W.; Chen, Y.; Xue, G.-R.; Yang, Q.; and Yu, Y. 2009. Translated learning: Transfer learning across different feature spaces. In *Advances in Neural Information Processing Systems 21*, 353–360.

Daumé III, H. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 256–263.

Duan, L.; Tsang, I. W.; Xu, D.; and Maybank, S. J. 2009. Domain transfer SVM for video concept detection. In *Proceedings of the 22nd IEEE Conference on Computer Vision and Pattern Recognition*, 1375–1381.

Duan, L.; Xu, D.; Tsang, I. W.; and Luo, J. 2010. Visual event recognition in videos by learning from Web data. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*, 1959–1966.

Duan, L.; Xu, D.; and Tsang, I. W. 2012. Learning with augmented features for heterogeneous domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning*, 711–718.

Futreal, P. A.; Coin, L.; Marshall, M.; Down, T.; Hubbard, T.; et al. 2004. A census of human cancer genes. *Nature Reviews Cancer* 4:177–183.

Gelfand, A. E., and Smith, A. F. M. 1990. Sampling-based approaches to calculating marginal densities. *Journal of the American Statistical Association* 85:398–409.

Gönen, M., and Alpaydın, E. 2011. Multiple kernel learning algorithms. *Journal of Machine Learning Research* 12(Jul):2211–2268.

Gönen, M. 2012. Bayesian efficient multiple kernel learning. In *Proceedings of the 29th International Conference on Machine Learning*, 1–8.

Gong, B.; Shi, Y.; Sha, F.; and Grauman, K. 2012. Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition*, 2066–2073.

Gopalan, R.; Li, R.; and Chellappa, R. 2011. Domain adaptation for object recognition: An unsupervised approach. In *Proceedings of the 13th IEEE International Conference on Computer Vision*, 999–1006.

Griffin, G.; Holub, A.; and Perona, P. 2007. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology.

Han, S.; Liao, X.; and Carin, L. 2012. Cross-domain multitask learning with latent probit models. In *Proceedings of the 29th International Conference on Machine Learning*, 73–80.

Hoffman, J.; Rodner, E.; Donahue, J.; Darrell, T.; and Saenko, K. 2013. Efficient learning of domain-invariant image representations. In *Proceedings of the 1st International Conference on Learning Representations*.

Jiang, W.; Zavesky, E.; Chang, S.-F.; and Loui, A. 2008. Cross-domain learning methods for high-level visual concept classification. In *Proceedings of the 15th IEEE International Conference on Image Processing*, 161–164.

Kulis, B.; Saenko, K.; and Darrell, T. 2011. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proceedings of the 24th IEEE Conference on Computer Vision and Pattern Recognition*, 1785–1792.

Lawrence, N. D., and Jordan, M. I. 2005. Semi-supervised learning via Gaussian processes. In *Advances in Neural Information Processing Systems 17*, 753–760.

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10):1345–1359.

Pan, S. J.; Tsang, I. W.; Kwok, J. T.; and Yang, Q. 2011. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks* 22(2):199–210.

Saenko, K.; Kulis, B.; Fritz, M.; and Darrell, T. 2010. Adapting visual category models to new domains. In *Proceedings of the 11th European Conference on Computer Vision, IV*, 213–226.

Schölkopf, B., and Smola, A. J. 2002. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press.

Shawe-Taylor, J., and Cristianini, N. 2004. *Kernel Methods for Pattern Analysis*. New York, NY, USA: Cambridge University Press.

TCGA Research Network. 2008. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature* 455:1061–1068.

Tipping, M. E. 2001. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research* 1:211–244.

Vapnik, V. N. 1998. *Statistical Learning Theory*. New York, NY, USA: John Wiley & Sons.