Stochastic Adversarial Learning for Domain Adaptation

Jen-Tzung Chien
Department of Electrical and Computer Engineering
National Chiao Tung University
Hsinchu, Taiwan

Ching-Wei Huang
Department of Electrical and Computer Engineering
National Chiao Tung University
Hsinchu, Taiwan

***Abstract*—Learning across domains is challenging especially when test data in target domain are sparse, heterogeneous and unlabeled. This challenge is even severe when building a deep stochastic neural model. This paper presents a stochastic semi-supervised learning for domain adaptation by using labeled data from source domain and unlabeled data from target domain. There are twofold novelties in the proposed method. First, a graphical model is constructed to identify the random latent features for classes as well as domains which are learned by variational inference. Second, we learn the class features which are discriminative among classes and simultaneously invariant to both domains. An adversarial neural model is introduced to pursue domain invariance. The domain features are explicitly learned to purify the extraction of class features for an improved classification. The experiments on sentiment classification illustrate the merits of the proposed stochastic adversarial domain adaptation.**

***Index Terms*—Adversarial learning, stochastic modeling, domain adaptation, sentiment classification**

I. Introduction

Domain adaptation aims to learn from a source data distribution to a different but related target data distribution which can achieve desirable performance in a target regression or classification task [1]–[7]. This issue is crucial for many natural language applications containing symbolic words or semantics, e.g. the spam filtering or the product review classification [8]. Such systems classify the emails or reviews for a target user or product by using the data distribution which is learned from those data originated from source user or product. In particular, we face the problem of transfer learning in presence of sparse, heterogeneous and unlabeled data in target domain. This problem is even more challenging when a deep neural model is constructed. This study presents a symbolic neural learning for feature-based approach to domain adaptation and sentiment classification. We learn a deep latent feature model where the learned features are invariant to domains so that the classification model trained from the features of source domain is adapted to target domain.

In the literature, the maximum mean discrepancy (MMD) [9]–[11] was proposed to measure the difference between two distributions based on a non-parametric kernel method. This MMD was minimized to train the latent features which were invariant to the migration from source domain to target domain. By incorporating the class labels, the estimated features are discriminative among classes. In [12], a multi-view and multi-objective learning were proposed to build semi-supervised model where feature extraction and pattern classifier were jointly optimized. In [13], the distribution matching for domain adaptation was realized through an adversarial neural network [14]–[20] which consisted of a feature extractor G_f and a pattern classifier G_y. A discriminator D was introduced to distinguish whether the estimated latents features belong to source domain or target domain. D, G_f and G_y were jointly trained to conduct the distribution matching according to a minimax two-player game theory. In [21], a variational fair autoencoder was proposed to learn a fair feature representation where a variational autoencoder (VAE) [22] was introduced to encourage independence between latent factors of variations existing in the observations $\mathbf{x}$. MMD measure was incorporated to optimize the independence. Traditionally, the latent features $\mathbf{z}_y$ of class labels $\mathbf{y}$ are extracted either by adversarial net or MMD method. The estimated class features are mixed with domain information which will deteriorate the classification performance.

This paper presents a stochastic adversarial classification network for domain adaptation by using the labeled data in source domain and the unlabeled data in target domain. A probabilistic semi-supervised model is proposed to characterize the sophisticated and heterogeneous relations between observations and latent features where *labels* $\mathbf{y}$ as well as *domains* $\mathbf{d}$ of observations $\mathbf{x}$ are represented. The distributions of the associated latent features $\mathbf{z}_y$ and $\mathbf{z}_d$ are driven by a stochastic neural network motivated by VAE. Distribution of these encoded features can be used for data generation. The variational inference procedure is implemented to construct a latent variable model which faithfully reflects the stochastic behavior of latent variables for domain adaptation. A variational lower bound of log likelihood, approximated by the stochastic gradient variational Bayes (SGVB) [22], is maximized. In particular, we propose two approaches to improve the classification performance based on this stochastic neural model. First, an adversarial neural network is merged to estimate data distributions which are invariant to different domains. A discriminator is optimized to maximize the ambiguity for classifying the features of source and target domains. Second, the domain features are explicitly characterized to increase the evidence of the estimated class features for

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

classification system. This method is evaluated by a domain adaptation task for sentiment classification.

II. Domain Adaptation

Assume that training samples are collected in source domain and target domain $\mathbf{d} = \{s, t\}$. Let $\{X^s, Y^s\} = \{(\mathbf{x}_1^s, \mathbf{y}_1^s), \ldots, (\mathbf{x}_n^s, \mathbf{y}_n^s)\}$ denote n labeled samples in source domain s. Here, $\mathbf{x}_i^s$ means the i^{th} training vector and $\mathbf{y}_i^s$ corresponds to its label vector. In addition, we have m unlabeled samples $X^t = \{\mathbf{x}_1^t, \ldots, \mathbf{x}_m^t\}$ from target domain t where the label information Y^t is missing. These two domains are related but not identical. The joint distributions $p(X^s, Y^s)$ and $p(X^t, Y^t)$ are different. Domain adaptation is a kind of transfer learning where marginal distributions $p(X^s)$ and $p(X^t)$ are different and conditional distributions of finding labels from data $p(Y^s|X^s)$ and $p(Y^t|X^t)$ should be identical.

A. Distribution matching

We first survey two related approaches to distribution matching for domain adaptation which can compensate the covariate shift between $p(X^s)$ and $p(X^t)$. The first one is to calculate the MMD measure [9] which is referred as a divergence between distributions of two data sets $\{X^s, X^t\}$ in a reproducing kernel Hilbert space $\mathcal{H}$

$$\text{MMD}(X^s, X^t) = \left\| \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{\phi}(\mathbf{x}_i^s) - \frac{1}{m} \sum_{j=1}^{m} \boldsymbol{\phi}(\mathbf{x}_j^t) \right\|_{\mathcal{H}} \quad (1)$$

where $\boldsymbol{\phi}(\cdot)$ denotes a basis function vector. MMD was estimated by using the Gaussian kernel and then minimized to pursue the distribution matching via re-weighting the instances in source domain [23]. MMD was also employed in construction of domain-invariant feature space [24], [25].

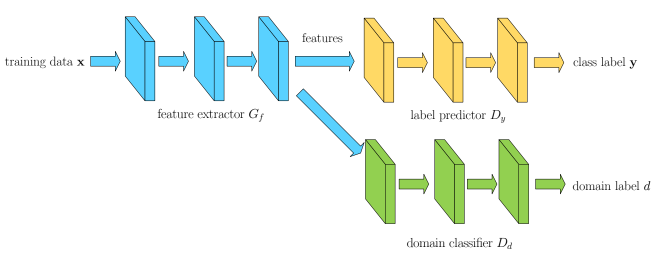


Fig. 1. Adversarial neural network for domain adaptation.

An alternative solution to distribution matching was developed by using an adversarial neural network (ANN) [13]. There are three components in ANN feedforward architecture; feature extractor G_f, label predictor D_y and domain classifier D_d. As illustrated in Figure 1, the features, extracted by G_f, are forwarded to find class label $\mathbf{y}$ of training sample $\mathbf{x}$ using label predictor D_y. Importantly, we estimate the domain-invariant features to pursue invariant distributions for source and target domains. A domain classifier D_d is applied to find domain $\mathbf{d}$ of a feature sample. The "confusion" in domain classification is maximized to assure invariance. The parameters of ANN are estimated via a minimax learning of latent features in G_f where the classification errors of labels in D_y are minimized and simultaneously the classification errors of domains in D_d are maximized.

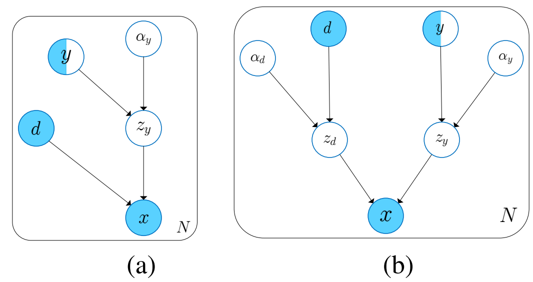


Fig. 2. Graphical models for (a) variational fair autoencoder and (b) the proposed variational domain and class representation.

B. Variational fair autoencoder

In [21], a variational fair autoencoder (VFA) was proposed to build a latent variable model for domain adaptation. VFA aims at learning a "fair" feature representations that are invariant to noise or sensitive factors which are not related to label. Figure 2(a) shows the graphical model of VFA which is seen as a semi-supervised model [26] where label information $\mathbf{y}$ is only available in source domain s. Following the property of variational autoencoder [22], the latent variable $\mathbf{z}_y$ of an input data $\mathbf{x}$ in VFA was driven by a posterior distribution or variational distribution $q(\mathbf{z}_y|\mathbf{y})$ based on an autoencoder. $\boldsymbol{\alpha}_y$ denotes the parameters of latent variable $\mathbf{z}_y$. Stochastic information of latent variable was characterized. The intractable problem in variational inference procedure was tackled by SGVB estimator where the expectation function in variational lower bound was approximated by sampling latent variable via a differentiable transformation with a noise variable. This yielded a simple differentiable unbiased estimator of lower bound. An analytical solution was therefore obtained to implement VFA through an error backpropagation algorithm. In general, an observed sample $\mathbf{x}$ is generated by the sensitive variable in applied domain $\mathbf{d}$ and the latent feature $\mathbf{z}_y$ with variation in class label $\mathbf{y}$ represented by $\boldsymbol{\alpha}_y$. In [26], the labels $\mathbf{y}$ of unlabeled data were treated as random. An additional term of classification error of unlabeled data was incorporated in deep generative model to ensure that the predictive posterior $q(\mathbf{z}_y|\mathbf{y})$ learns from both labeled and unlabeled data. This enriched the latent feature representation of class label $\mathbf{z}_y$.

Nevertheless, the richness of this latent variable model was constrained because the class feature $\mathbf{z}_y$ is still contaminated with noise or domain factors which will deteriorate the classification performance. For example, in the task of Amazon review with classes or ratings of positive and negative. We build a model adapting from source domain "Electronics" to target domain "Game". This model may be learned to catch the features or semantics for words 'compact' in "Electronics" and 'hooked' in "Games" and those for many other words corresponding to two classes. Class feature $\mathbf{z}_y$ does vary by domains. Furthermore, the features of domain words, e.g.

camera, phone and TV, in "Electronics" do contain variations. It is crucial to characterize these variations to elevate the performance of classification system.

III. Stochastic Adversarial Learning

This paper presents a variational and adversarial learning for latent feature representation.

A. Model construction

As shown in Figure 2(b), latent features of labels $\mathbf{y}$ as well as domains $\mathbf{d}$ are explicitly expressed and learned to build a *domain-invariant feature space* for domain adaptation. The domain variations are separately modeled to prevent leakage of domain factor $\mathbf{z}_d$ into the extraction of class feature $\mathbf{z}_y$. We would like to maximally correlate the class feature $\mathbf{z}_y$ with class label $\mathbf{y}$ and impose $\mathbf{z}_y$ to be invariant to the change of domain $\mathbf{d}$. Similarly, the domain feature $\mathbf{z}_d$ is identified by maximally correlating with domain label $\mathbf{d}$ and making invariance with class label $\mathbf{y}$. Separating the parameter $\boldsymbol{\alpha}_d$ of domain feature $\mathbf{z}_d$ from that $\boldsymbol{\alpha}_y$ of class feature $\mathbf{z}_y$ can help finding a "purified" class feature $\mathbf{z}_y$ to improve classification. Without loss of generality, we present a variational domain and class (VDC) representation for domain adaptation. There are twofold extensions in this study. First, the variational inference is implemented to learn the distributions of latent features which allow data reconstruction for deep generative model. Second, an adversarial neural network is merged to achieve the matching of variational distributions of class features between source and target domains.

Variational autoencoder [22] is introduced to infer the proposed VDC model. An encoder using variational posterior $q_{\boldsymbol{\phi}}(\mathbf{z}_y, \mathbf{z}_d, \boldsymbol{\alpha}_y, \boldsymbol{\alpha}_d, \mathbf{y}|\mathbf{x}, \mathbf{d})$ with variational parameter $\boldsymbol{\phi}$ and a decoder using generative likelihood $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_y, \mathbf{z}_d, \mathbf{y}, \mathbf{d}, \boldsymbol{\alpha}_y, \boldsymbol{\alpha}_d)$ with model parameter $\boldsymbol{\theta}$ are merged in inference of an integrated deep neural network. Here, source domain s and target domain t are denoted by a domain vector $\mathbf{d}$ as $[1\ \ 0]^\top$ and $[0\ \ 1]^\top$, respectively. Variational posterior $q_\phi(\mathbf{z}_y, \mathbf{z}_d, \boldsymbol{\alpha}_y, \boldsymbol{\alpha}_d, \mathbf{y}|\mathbf{x}, \mathbf{d})$ is used to approximate the true posterior $p_{\boldsymbol{\theta}}(\mathbf{z}_y, \mathbf{z}_d, \boldsymbol{\alpha}_y, \boldsymbol{\alpha}_d, \mathbf{y}|\mathbf{x}, \mathbf{d})$ in variational inference. The factorizations of decoder $p(\mathbf{x}, \mathbf{z}_y, \mathbf{z}_d, \mathbf{y}, \mathbf{d}, \boldsymbol{\alpha}_y, \boldsymbol{\alpha}_d)$ and encoder $q(\mathbf{z}_y, \mathbf{z}_d, \boldsymbol{\alpha}_y, \boldsymbol{\alpha}_d, \mathbf{y}|\mathbf{x}, \mathbf{d})$ are expressed by $p(\mathbf{x}|\mathbf{z}_y, \mathbf{z}_d)p(\mathbf{z}_y|\mathbf{y}, \boldsymbol{\alpha}_y)p(\mathbf{z}_d|\mathbf{d}, \boldsymbol{\alpha}_d)p(\mathbf{y})p(\boldsymbol{\alpha}_y)p(\boldsymbol{\alpha}_d)$ and $q(\mathbf{z}_y|\mathbf{x}, \mathbf{d})q(\mathbf{z}_d|\mathbf{x}, \mathbf{d})q(\boldsymbol{\alpha}_y|\mathbf{z}_y, \mathbf{y})q(\boldsymbol{\alpha}_d|\mathbf{z}_d, \mathbf{d})q(\mathbf{y}|\mathbf{z}_y)$, respectively. In case that class label $\mathbf{y}$ is unknown, $p(\mathbf{y})$ in decoder and $q(\mathbf{y}|\mathbf{z}_y)$ in encoder are disregarded. The factorized distributions of real-valued variables and discrete-valued variables in $p_{\boldsymbol{\theta}}(\cdot)$ and $q_{\boldsymbol{\phi}}(\cdot)$ are represented by Gaussian distribution $\mathcal{N}(\cdot)$ and category (multinomial) distribution $\text{Cat}(\cdot)$, respectively, given by

$$
\begin{aligned}
&p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_y, \mathbf{z}_d) = f_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_y, \mathbf{z}_d),\ p(\mathbf{y}) = \text{Cat}(\mathbf{y}|\boldsymbol{\pi}_0) \\
&p_{\boldsymbol{\theta}}(\mathbf{z}_y|\mathbf{y}, \boldsymbol{\alpha}_y) = \mathcal{N}(\mathbf{z}_y|\boldsymbol{\mu} = f_{\boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\alpha}_y), \boldsymbol{\sigma} = e^{f_{\boldsymbol{\theta}}(\mathbf{y}, \boldsymbol{\alpha}_y)}) \\
&p_{\boldsymbol{\theta}}(\mathbf{z}_d|\mathbf{d}, \boldsymbol{\alpha}_d) = \mathcal{N}(\mathbf{z}_d|\boldsymbol{\mu} = f_{\boldsymbol{\theta}}(\mathbf{d}, \boldsymbol{\alpha}_d), \boldsymbol{\sigma} = e^{f_{\boldsymbol{\theta}}(\mathbf{d}, \boldsymbol{\alpha}_d)}) \\
&p(\boldsymbol{\alpha}_y) = \mathcal{N}(\boldsymbol{\alpha}_y|\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0),\ p(\boldsymbol{\alpha}_d) = \mathcal{N}(\boldsymbol{\alpha}_d|\boldsymbol{\mu}_0, \boldsymbol{\sigma}_0)
\end{aligned}
\tag{2}
$$

and

$$
\begin{aligned}
&q_{\boldsymbol{\phi}}(\mathbf{z}_y|\mathbf{x}, \mathbf{d}) = \mathcal{N}(\mathbf{z}_y|\boldsymbol{\mu} = f_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{d}), \boldsymbol{\sigma} = e^{f_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{d})}) \\
&q_{\boldsymbol{\phi}}(\mathbf{z}_d|\mathbf{x}, \mathbf{d}) = \mathcal{N}(\mathbf{z}_d|\boldsymbol{\mu} = \tilde{f}_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{d}), \boldsymbol{\sigma} = e^{\tilde{f}_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{d})}) \\
&q_{\boldsymbol{\phi}}(\boldsymbol{\alpha}_y|\mathbf{z}_y, \mathbf{y}) = \mathcal{N}(\boldsymbol{\alpha}_y|\boldsymbol{\mu} = f_{\boldsymbol{\phi}}(\mathbf{z}_y, \mathbf{y}), \boldsymbol{\sigma} = e^{f_{\boldsymbol{\phi}}(\mathbf{z}_y, \mathbf{y})}) \\
&q_{\boldsymbol{\phi}}(\boldsymbol{\alpha}_d|\mathbf{z}_d, \mathbf{d}) = \mathcal{N}(\boldsymbol{\alpha}_d|\boldsymbol{\mu} = f_{\boldsymbol{\phi}}(\mathbf{z}_d, \mathbf{d}), \boldsymbol{\sigma} = e^{f_{\boldsymbol{\phi}}(\mathbf{z}_d, \mathbf{d})}) \\
&q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{z}_y) = \text{Cat}(\mathbf{y}|\boldsymbol{\pi} = f_{\boldsymbol{\phi}}(\mathbf{z}_y))
\end{aligned}
\tag{3}
$$

where $f_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_y, \mathbf{z}_d)$ is an appropriate data distribution which is an Gaussian in this study. Mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}$ in $p_{\boldsymbol{\theta}}(\cdot)$ and $q_{\boldsymbol{\phi}}(\cdot)$ are expressed by functions $f_{\boldsymbol{\theta}}(\cdot)$ and $f_{\boldsymbol{\phi}}(\cdot)$, respectively, which are estimated by using different neural networks. Latent variables in VDC consist of $\{\mathbf{z}_y, \mathbf{z}_d, \boldsymbol{\alpha}_y, \boldsymbol{\alpha}_d, \mathbf{y}\}$ with class $\mathbf{z}_y$ and domain features $\mathbf{z}_d$.

B. Variational lower bound

In stochastic learning or variational inference of VDC, we maximize the logarithm of marginal likelihood by using N i.i.d. training vectors $\log p(\mathbf{x}_i, \cdots, \mathbf{x}_N) = \sum_{i=1}^N \log p(\mathbf{x}_i)$ where log likelihood $\log p(\mathbf{x}_i)$ is yielded by

$$
\begin{aligned}
&\text{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_{yi}, \mathbf{z}_{di}, \boldsymbol{\alpha}_{yi}, \boldsymbol{\alpha}_{di}|\mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i))\| \\
&\qquad p_{\boldsymbol{\theta}}(\mathbf{z}_{yi}, \mathbf{z}_{di}, \boldsymbol{\alpha}_{yi}, \boldsymbol{\alpha}_{di}|\mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i)) \\
&\quad + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i).
\end{aligned}
\tag{4}
$$

In right-hand-side (RHS) of Eq. 4, the first term is the Kullback-Leiblier (KL) divergence between variational posterior $q_{\boldsymbol{\phi}}(\cdot)$ and true posterior $p_{\boldsymbol{\theta}}(\cdot)$ and the second term $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i)$ denotes the variational lower bound of log likelihood of i-th sample which is obtained by RHS of the following inequality

$$
\begin{aligned}
\log p(\mathbf{x}_i) \geq\ & \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_{yi}, \mathbf{z}_{di}, \boldsymbol{\alpha}_{yi}, \boldsymbol{\alpha}_{di}|\mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i)} \\
&[-\log q_{\boldsymbol{\phi}}(\mathbf{z}_{yi}, \mathbf{z}_{di}, \boldsymbol{\alpha}_{yi}, \boldsymbol{\alpha}_{di}|\mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i) \\
&+ \log p_{\boldsymbol{\theta}}(\mathbf{z}_{yi}, \mathbf{z}_{di}, \boldsymbol{\alpha}_{yi}, \boldsymbol{\alpha}_{di}|\mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i)].
\end{aligned}
\tag{5}
$$

VDC model is inferred by maximizing this lower bound with respect to variational parameters $\boldsymbol{\phi}$ and model parameters $\boldsymbol{\theta}$. Lower bound for a sample is accordingly expanded as

$$
\begin{aligned}
&\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}, \mathbf{y}, \mathbf{d}) \\
&= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_y|\mathbf{x}, \mathbf{d})q_{\boldsymbol{\phi}}(\mathbf{z}_d|\mathbf{x}, \mathbf{d})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}_y, \mathbf{z}_d)] \\
&+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\alpha}_y|\mathbf{z}_y, \mathbf{y})q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{z}_y)} \\
&\qquad [-\text{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_y|\mathbf{x}, \mathbf{d})\|p_{\boldsymbol{\theta}}(\mathbf{z}_y|\mathbf{y}, \boldsymbol{\alpha}_y))] \\
&+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{\alpha}_d|\mathbf{z}_d, \mathbf{d})}[-\text{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}_d|\mathbf{x}, \mathbf{d})\|p_{\boldsymbol{\theta}}(\mathbf{z}_d|\mathbf{d}, \boldsymbol{\alpha}_d))] \\
&+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_y|\mathbf{x}, \mathbf{d})q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{z}_y)}[-\text{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\alpha}_y|\mathbf{z}_y, \mathbf{y})\|p(\boldsymbol{\alpha}_y))] \\
&+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_d|\mathbf{x}, \mathbf{d})}[-\text{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\alpha}_d|\mathbf{z}_d, \mathbf{d})\|p(\boldsymbol{\alpha}_d))] \\
&+ \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_y|\mathbf{x}, \mathbf{d})}[-\text{KL}(q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{z}_y)\|p(\mathbf{y})].
\end{aligned}
\tag{6}
$$

Index i is neglected for ease of expression. Notably, this bound is calculated by using the labeled data from source domain $\{\mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i = [1\ \ 0]^\top\}_{i=1}^n$ and the unlabeled data from target domain $\{\mathbf{x}_j, \mathbf{d}_j = [0\ \ 1]^\top\}_{j=1}^m$. Lower bound $\mathcal{L}(\cdot)$ is either from source domain $\mathcal{L}_s(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i)$ or from target domain $\mathcal{L}_t(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}_j, \mathbf{d}_j)$. In addition, we also maximize an entropy term $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_{yi}|\mathbf{x}_i, \mathbf{d}_i)}[-\log q_{\boldsymbol{\phi}}(\mathbf{y}_i|\mathbf{z}_{yi})]$ in objective function to assure the predictive posterior $q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{z}_y)$ learned from both labeled and unlabeled data. The objective function $\mathcal{F}_{\text{VDC}}(\boldsymbol{\theta}, \phi; \mathbf{X}, \mathbf{Y}, \mathbf{d})$ is constructed by

$$\begin{aligned} &\textstyle\sum_{i=1}^{n} \mathcal{L}_s(\boldsymbol{\theta}, \phi; \mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i) + \sum_{j=1}^{m} \mathcal{L}_t(\boldsymbol{\theta}, \phi; \mathbf{x}_j, \mathbf{d}_j) \\ &\quad + \lambda \textstyle\sum_{i=1}^{n} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_{yi}|\mathbf{x}_i, \mathbf{d}_i)}[-\log q_{\boldsymbol{\phi}}(\mathbf{y}_i|\mathbf{z}_{yi})] \end{aligned} \tag{7}$$

using training data

$$\{\mathbf{X}, \mathbf{Y}, \mathbf{d}\} = \{X^s, X^t, Y^s, \mathbf{d}\}. \tag{8}$$

λ is a regularization parameter. $\mathcal{L}_t(\cdot)$ is formed by $\mathcal{L}_s(\cdot)$ with the last term in RHS of Eq. (6). A latent domain and class representation is finally implemented.

C. Adversarial learning

The distribution matching based on adversarial learning is further incorporated into VDC model to improve domain adaptation. As a result, the distributions of class features $\mathbf{z}_y$ are fitted to both source and target domains. Different from [13], an adversarial neural network (ANN) is implemented to evaluate the hybrid feature space $\{\mathbf{z}_d, \mathbf{z}_y\}$ which is constructed for variational domain and class (VDC) representation. This evaluation is performed via an adversarial process which maximizes the ambiguity of latent class features $\mathbf{z}_y$ between source domain and target domain. The resulting solution is hereafter called the Variational and Adversarial learning for Domains and Classes (VADC). To fulfill VADC framework, a discriminator based on neural network $D = f_{\boldsymbol{\varphi}}(\mathbf{z}_y)$ is additionally introduced to judge whether the class feature $\mathbf{z}_y$ of an observation $\mathbf{x}_i$ or $\mathbf{x}_j$ are extracted from source domain $\mathbf{d}_i = [1\ 0]^\top$ or target domain $\mathbf{d}_j = [0\ 1]^\top$. Importantly, we maximize the ambiguity or equivalently "minimize" the *negative cross entropy error* function between discriminator outputs $\{f_{\boldsymbol{\varphi}}(\mathbf{z}_{yi})\}_{i=1}^{n+m}$ and desirable outputs $\{\mathbf{d}_i\}_{i=1}^{n+m}$ over observations in both domains $\{\mathbf{x}_i\}_{i=1}^{n+m}$. Discriminator output is seen as the class posterior $f_{\boldsymbol{\varphi}}(\mathbf{z}_{yi}) = p(\mathbf{d}|\mathbf{z}_{yi}, \boldsymbol{\varphi})$. This VADC model is inferred through a minimax optimization where a generative model G with parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ based on VDC and a discriminative model D with parameter $\boldsymbol{\varphi}$ based on ANN are jointly trained. The optimization problem is correspondingly formed by

$$\max_{\boldsymbol{\phi}, \boldsymbol{\theta}} \min_{\boldsymbol{\varphi}} \mathcal{F}_{\text{VADC}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\varphi}; \mathbf{X}, \mathbf{Y}, \mathbf{d}) \tag{9}$$

using the objective $\mathcal{F}_{\text{VADC}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\varphi}; \mathbf{X}, \mathbf{Y}, \mathbf{d})$ formulated by

$$\begin{aligned} &\textstyle\sum_{i=1}^{n} \mathcal{L}_s(\boldsymbol{\theta}, \phi; \mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i) + \sum_{j=1}^{m} \mathcal{L}_t(\boldsymbol{\theta}, \phi; \mathbf{x}_j, \mathbf{d}_j) \\ &\quad + \lambda_1 \textstyle\sum_{i=1}^{n} \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}_{yi}|\mathbf{x}_i, \mathbf{d}_i)}[-\log q_{\boldsymbol{\phi}}(\mathbf{y}_i|\mathbf{z}_{yi})] \\ &\quad + \lambda_2 \textstyle\sum_{i=1}^{n+m} \sum_c d_{ic} f_{\boldsymbol{\varphi}}(z_{yic}) \end{aligned} \tag{10}$$

where $\mathbf{d}_i = \{d_{ic}\}$ and $\mathbf{z}_{yi} = \{z_{yic}\}$ with domain index c. The last term in Eq. (10) corresponds to the negative cross entropy error function. Therefore, using this integrated objective, we can learn a variational and adversarial model for domain adaptation where the likelihood of generator in Figure 2(b) and the entropy of posterior predictor $q_{\boldsymbol{\phi}}(\mathbf{y}|\mathbf{z}_y)$ with parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$ are maximized subject to the condition that the negative cross entropy error function of discriminator $f_{\boldsymbol{\varphi}}(\mathbf{z}_y)$ with parameter $\boldsymbol{\varphi}$ is minimized. The regularization parameters λ_1 for maximum entropy and λ_2 for adversarial learning are adopted to balance the tradeoff among these three factors.

D. Implementation issue

In the inference procedure, the expectation terms in objective function of VDC or VADC and their derivatives are intractable. To deal with this issue, we apply SGVB estimator [22] and approximate the expectation through the sampling of latent variables $\{\mathbf{z}_y, \mathbf{z}_d, \boldsymbol{\alpha}_y, \boldsymbol{\alpha}_d, \mathbf{y}\}$. A re-parameterization trick is employed to avoid high variance in sampling procedure. Accordingly, we first re-parameterize a latent variable z or α using a differentiable transformation given by an auxiliary noise variable ϵ or ζ. Transformations of real-valued variables $\{\mathbf{z}_y, \mathbf{z}_d, \boldsymbol{\alpha}_y, \boldsymbol{\alpha}_d\}$ and discrete-valued variable $\mathbf{y}$ are described as

$$\begin{aligned} &\mathbf{z}_y = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_y \quad \text{where} \quad \boldsymbol{\epsilon}_y \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ &\quad \boldsymbol{\mu} = f_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{d}),\ \boldsymbol{\sigma} = \exp(f_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{d})) \\ &\mathbf{z}_d = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon}_d \quad \text{where} \quad \boldsymbol{\epsilon}_d \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ &\quad \boldsymbol{\mu} = \tilde{f}_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{d}),\ \boldsymbol{\sigma} = \exp(\tilde{f}_{\boldsymbol{\phi}}(\mathbf{x}, \mathbf{d})) \\ &\boldsymbol{\alpha}_y = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\zeta}_y \quad \text{where} \quad \boldsymbol{\zeta}_y \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ &\quad \boldsymbol{\mu} = f_{\boldsymbol{\phi}}(\mathbf{z}_y, \mathbf{y}),\ \boldsymbol{\sigma} = \exp(f_{\boldsymbol{\phi}}(\mathbf{z}_y, \mathbf{y})) \\ &\boldsymbol{\alpha}_d = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\zeta}_d \quad \text{where} \quad \boldsymbol{\zeta}_d \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \\ &\quad \boldsymbol{\mu} = f_{\boldsymbol{\phi}}(\mathbf{z}_d, \mathbf{d}),\ \boldsymbol{\sigma} = \exp(f_{\boldsymbol{\phi}}(\mathbf{z}_d, \mathbf{d})) \\ &\mathbf{y} = g(\log(\boldsymbol{\pi} + \mathbf{c}) + \boldsymbol{\xi}) \end{aligned} \tag{11}$$

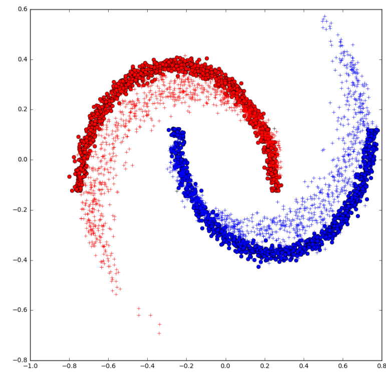
where $\boldsymbol{\pi} = f_{\boldsymbol{\phi}}(\mathbf{z}_y)$, $\mathbf{c}$ is fixed and $\boldsymbol{\xi}$ is sampled from a standard Gumbel distribution. $g(\cdot)$ is a function that assigns 1 to the entry with the largest value and 0 to the other entries. These transformations are used to approximate the expectations in objective function by Monte Carlo estimates. Notably, the Gaussian parameters $\{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ are estimated from the outputs of neural networks $f_{\boldsymbol{\phi}}(\cdot)$ with parameters $\boldsymbol{\phi}$ by using the inputs $\{\mathbf{x}, \mathbf{d}\}$ for latent features $\{\mathbf{z}_y, \mathbf{z}_d\}$ and the inputs $\{\mathbf{z}_y, \mathbf{y}, \mathbf{z}_d, \mathbf{d}\}$ for latent variables $\{\boldsymbol{\alpha}_y, \boldsymbol{\alpha}_d\}$. SGVB estimator is implemented by maximizing for generator and minimizing for discriminator via

$$\begin{aligned} &\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta \nabla_{\boldsymbol{\theta}} \mathcal{F}_{\text{VADC}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\varphi}; \mathbf{X}, \mathbf{Y}, \mathbf{d}) \\ &\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} + \eta \nabla_{\boldsymbol{\phi}} \mathcal{F}_{\text{VADC}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\varphi}; \mathbf{X}, \mathbf{Y}, \mathbf{d}) \\ &\boldsymbol{\varphi} \leftarrow \boldsymbol{\varphi} - \eta \nabla_{\boldsymbol{\varphi}} \mathcal{F}_{\text{VADC}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\varphi}; \mathbf{X}, \mathbf{Y}, \mathbf{d}) \end{aligned} \tag{12}$$

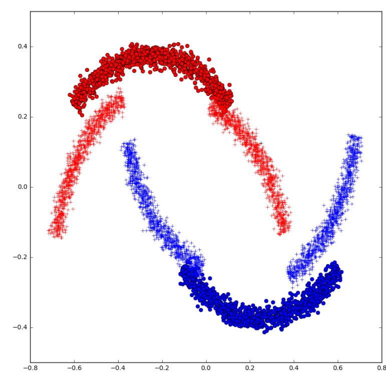
where η is learning rate. In the implementation, the discriminator $D = f_{\boldsymbol{\varphi}}(\mathbf{z}_y)$ is optimized with K updating steps before one step of updating for optimization of parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$ for generative model G [14]. This trick tends to maintain the estimated discriminator D near its optimal solution provided that the generator G changes slowly. In case that the discriminator D is optimized to completion before updating the generator G with one step, the over-fitting problem will happen too early in presence of a limited size of training data. Algorithm 1 shows the stochastic training procedure in VADC where the discriminator is updated k steps before updating the generator. k and L are hyperparameters.

Algorithm 1: Stochastic gradient descent training algorithm for VADC

$\{\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\varphi}\} \leftarrow$ initialize parameters
for *number of training iterations* **do**
  Calculation for discriminator
  for k *steps* **do**
    $\{\mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i\}_{i=1}^{n'+m'} \leftarrow$ sample minibatch of $n' + m'$ datapoints from $\{\mathbf{X}, \mathbf{Y}, \mathbf{d}\}$
    $\{\boldsymbol{\epsilon}^{(l)}\}_{l=1}^{L} \leftarrow$ get L random samples from noise distribution
    $\{\mathbf{z}_{yi}^{(l)}\}_{i=1,l=1}^{n'+m',L} \leftarrow$ get samples of latent variable
    $f_{\boldsymbol{\varphi}}(\mathbf{z}_{yi}^{(l)}) \leftarrow$ get the discriminator output
    $\mathbf{g}_{\boldsymbol{\varphi}} \leftarrow -\nabla_{\boldsymbol{\varphi}} \mathcal{F}_{\text{VADC}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\varphi}; \mathbf{X}, \mathbf{Y}, \mathbf{d})$
    $\boldsymbol{\varphi} \leftarrow$ update parameter using gradient $\mathbf{g}_{\boldsymbol{\varphi}}$
  end
  Calculation for generator
  $\{\mathbf{x}_i, \mathbf{y}_i, \mathbf{d}_i\}_{i=1}^{n'+m'} \leftarrow$ sample minibatch of $n' + m'$ datapoints from $\{\mathbf{X}, \mathbf{Y}, \mathbf{d}\}$
  $\{\boldsymbol{\epsilon}^{(l)}, \boldsymbol{\zeta}^{(l)}, \boldsymbol{\xi}^{(l)}\}_{l=1}^{L} \leftarrow$ get L random samples from noise distribution
  $\{\mathbf{z}_{yi}^{(l)}, \mathbf{z}_{di}^{(l)}, \boldsymbol{\alpha}_{yi}^{(l)}, \boldsymbol{\alpha}_{di}^{(l)}, \mathbf{y}_i^{(l)}\}_{i=1,l=1}^{n'+m',L} \leftarrow$ get samples of latent variable
  $f_{\boldsymbol{\varphi}}(\mathbf{z}_{yi}^{(l)}) \leftarrow$ get the discriminator output
  $\{\mathbf{g}_{\boldsymbol{\theta}}, \mathbf{g}_{\boldsymbol{\phi}}\} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \mathcal{F}_{\text{VADC}}(\boldsymbol{\theta}, \boldsymbol{\phi}, \boldsymbol{\varphi}; \mathbf{X}, \mathbf{Y}, \mathbf{d})$
  $\{\boldsymbol{\theta}, \boldsymbol{\phi}\} \leftarrow$ update parameters using gradients $\{\mathbf{g}_{\boldsymbol{\theta}}, \mathbf{g}_{\boldsymbol{\phi}}\}$
end



(a)



(b)

Fig. 3. Twin-moon synthetic data for (a) Condition A and (b) Condition B. Color refers to class label. The samples marked by solid $\circ$ are data from source domain while the samples marked by $+$ are data from target domain.

IV. Experiments

A series of experiments are conducted for domain adaptation which includes a synthetic task and a real-world task for sentiment classification.

A. Experimental setup

The first task is a binary classification on two-dimensional twin-moon synthetic data in presence of two classes, upper moon and lower moon, with source domain marked by solid $\circ$ and target domain marked by $+$. Radius of moon is 0.5. There are two experimental conditions (A and B) in this evaluation. Figure 3(a) shows the Condition A that the data in target domain are rotated by an angle which is Gaussian distributed with mean $\pi/8$ and variance $\pi/80$. Figure 3(b) illustrates the Condition B that data in both domains are sampled from different shifted and overlapped segments. Obviously, the domain variation and the classification ambiguity in Condition B are more severe than those in Condition A. We would like to evaluate different methods based on these two conditions. For each condition, there were 2K samples in source domain with class labels and 2K samples in target domains without class labels. An additional set of 400 samples from individual domains was collected as test data.

The second task is developed for sentiment classification by using the multi-domain sentiment dataset [27], [28] which contains Amazon product reviews on four products including kitchen appliances, DVDs, books, electronics. Each product is seen as a domain. The goal is to classify the review into positive or negative reviews. In training session, there are 1000 positive reviews (higher than 3 stars) and 1000 negative reviews (lower than 3 stars) on each product or domain. We train a binary classifier from labeled reviews in source domain and unlabeled reviews in target domain and use it to predict whether a test review in target domain is positive or negative. The dictionary was built by top 2K frequent words. The tf-idf reweighting method was applied to obtain 2000-dimensional observation vector $\mathbf{x}$. Test data were composed of 500 positive reviews and 500 negative reviews. In these two tasks, 20% of training data were held out for validation to select regularization parameters $\{\lambda, \lambda_1, \lambda_2\}$ and the other hyperparameters.

In the experiments, the baseline system was built by neural network (NN) model with topology 2-10-5-2 for 1[st] task and 2000-500-50-2 for 2[nd] task by using labeled data from source domain. Two hidden layers with different number of neurons were considered. For comparison, the distribution matching methods using maximum mean discrepancy (MMD) and adversarial neural network (ANN) were implemented over the features in hidden layers by using data from both domains. The resulting methods, named by NN-MMD and NN-ANN [13], were carried out. Moreover, the variational fair autoencoder (VFA) was carried out for comparative study. In [21], VFA was proposed as a stand-alone method or a combined method with MMD (VFA-MMD). Data from both domains were used. In this study, we exploited a new VFA combined with ANN (VFA-ANN) which was implemented by introducing a discriminator to maximize the ambiguity of classifying the variational features $\mathbf{z}_y$ between source and target domains. For comparison, we implemented the proposed VDC and VADC where the variational domain and class features were learned. VADC was a realization of VDC-ANN where adversarial learning was performed in VDC representation. Interestingly, we could also implement a new realization VDC-MMD by adding the MMD term in a hybrid objective for VDC learning. In the experiments, we applied the random kitchen sinks to approximate MMD [29]. Adam algorithm [30] was used. Size of minibatch was 100. In implementation of VFA and VDC, all encoders and decoders were built by neural network with one hidden layer consisting of 10 neurons. There were nine blocks of neural networks in VDC which was seen as a deep model. In the 1$^{\text{st}}$ task, individual 10 neurons in hidden layers of encoder and decoder were specified. Using VFA, dimensions of $\mathbf{z}_y$ and $\boldsymbol{\alpha}_y$ were 10 and 5, respectively. Using VDC, dimensions of $\mathbf{z}_y$ and $\mathbf{z}_d$ were both 5 and those of $\boldsymbol{\alpha}_y$, $\boldsymbol{\alpha}_d$ were both 5. In the 2$^{\text{nd}}$ task, dimensions of $\mathbf{z}_y$, $\mathbf{z}_d$, $\boldsymbol{\alpha}_y$ and $\boldsymbol{\alpha}_d$ were all 50. Individual 200 neurons in hidden layer of encoder and decoder were used. In both tasks, the activation function was sigmoid, the step number $K = 10$ was set, the dimension of MMD approximator was 500 and the number of sample in Monte Carlo estimator was one. Different models were trained with convergence.

TABLE I
CLASSIFICATION ACCURACY (%) FOR ADAPTATION UNDER DIFFERENT CONDITIONS BY USING TWIN-MOON SYNTHETIC DATA.

	Condition A	Condition B
NN	84.3	60.4
NN-MMD	90.2	68.7
NN-ANN	90.9	73.5
VFA	87.9	68.5
VFA-MMD	94.5	74.8
VFA-ANN	94.9	77.5
VDC	88.3	74.7
VDC-MMD	95.1	79.0
VDC-ANN (VADC)	**96.1**	**82.5**

B. Experimental results

Table I compares the classification accuracies of different neural models by using twin-moon synthetic data under the Conditions A and B. In this comparison, we evaluate how different neural models, namely NN, VFA and the proposed VDC, perform for domain adaptation without and with distribution matching based on MMD and ANN. This binary classification is evaluated by changing the variations of data and their domains. Basically, the accuracies in Condition A are higher than those in Condition B because Condition B are more adverse than Condition A. Semi-supervised learning using VFA and VDC performs better than supervised learning using NN owing to twofold reasons. First, compared with NN, VFA and VDC are learned with additional unlabeled data from target domain. Second, stochastic learning in VFA and VDC provides better latent feature representation than deterministic modeling in NN. In addition, we find that the distribution matching consistently works for different models and conditions. ANN obtains improvement compared with MMD in Condition A. The improvement becomes significant in Condition B. In Condition A, VDC, VDC-MMD and VDC-ANN perform better than VFA, VFA-MMD and VFA-ANN, respectively. In Condition B, VDC related methods are *much better* than VFA related methods. This demonstrates that the latent domain and class representation in VDC does extract the purified and informative class features for improving the classification results. Among different methods, the best result in Condition B is obtained by VDC-ANN or equivalently VADC.

TABLE II
CLASSIFICATION ACCURACY (%) FOR ADAPTATION IN DIFFERENT DOMAINS (K: KITCHEN APPLIANCES, D: DVDS, B: BOOKS, E: ELECTRONICS)

	D→B	B→D	B→E
NN	74.2	77.2	70.3
NN-MMD	76.3	79.4	74.0
NN-ANN	77.1	80.7	74.1
VFA	76.3	77.1	72.5
VFA-MMD	78.2	80.0	75.1
VFA-ANN	78.9	81.1	76.5
VDC	76.9	77.5	72.9
VDC-MMD	79.1	80.9	75.5
VDC-ANN (VADC)	**80.5**	**82.9**	**77.9**
	E→K	**K→D**	**D→K**
NN	83.0	68.0	75.6
NN-MMD	84.2	72.8	80.4
NN-ANN	86.0	74.1	82.1
VFA	83.9	71.3	76.9
VFA-MMD	85.9	73.9	81.8
VFA-ANN	86.5	75.3	82.9
VDC	86.7	74.2	79.7
VDC-MMD	88.1	77.0	82.7
VDC-ANN (VADC)	**90.1**	**79.9**	**84.2**

Table II reports the performance of different methods for sentiment classification where adaptation among various domains is evaluated. Totally six pairs of domains are examined. The classification results indicate that applying distribution matching methods, MMD and ANN, consistently improves system performance. Stochastic learning using additional unlabeled data works well. In most cases, ANN performs better than MMD when combining with NN, VFA and VDC. But, ANN is more computationally demanding than MMD. In addition, the improvement of VDC methods over VFA methods is consistent in this comparison even when the domains of DVDs, Books and Electronics are relatively close and the reviews in these domains contain similar content. The improvement becomes significant when the pairs of adaptation domains, Electronics to Kitchen, Kitchen to DVDs and DVDs to Kitchen, are investigated. The variations of the domains in these three pairs are generally larger than those in the other three pairs. VDC is specialized to deal with the challenge of variations in different domains for domain adaptation.

V. Conclusions

We have presented a new latent variable model for domain adaptation based on variational and adversarial learning. This model run the stochastic learning for latent domain and class representation where latent features of domains and classes were separately characterized. Stochastic modeling of latent features was performed to reflect the essence of data generation or reconstruction. The classification system was benefited by using the enhanced class features. At the same time, the adversarial learning was performed to extract the class features which are invariant to different domains. A discriminator was introduced to maximize the ambiguity of classifying the estimated class features to source domain and target domain. An integrated objective learning was implemented in the experiments on using synthesis data and real-world data. The proposed method was improved especially for the cases of adaptation tasks in presence of high variations across domains.

References

[1] Sinno Jialin Pan, Ivor W. Tsang, James T. Kwok, and Qiang Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[2] Meina Kan, Shiguang Shan, and Xilin Chen, "Bi-shifting auto-encoder for unsupervised domain adaptation," in *Proc. of IEEE International Conference on Computer Vision*, 2015, pp. 3846–3854.

[3] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan, "Domain separation networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 343–351.

[4] Jen-Chieh Tsai and Jen-Tzung Chien, "Adversarial domain separation and adaptation," in *Prof. of IEEE International Workshop on Machine Learning for Signal Processing*, 2017, pp. 1–6.

[5] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I. Jordan, "Partial transfer learning with selective adversarial networks," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 2724–2732.

[6] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu, "Wasserstein distance guided representation learning for domain adaptation," in *Proc. of AAAI Conference on Artificial Intelligence*, 2018, pp. 4058–4065.

[7] Jen-Tzung Chien and Yu-Ying Lyu, "Partially adversarial learning and adaptation," in *Proc. of European Signal Processing Conference*, 2019, pp. 1444–1448.

[8] Jing Jiang and Cheng-Xiang Zhai, "Instance weighting for domain adaptation in NLP," in *Proc. of Annual Meeting of the Association of Computational Linguistics*, 2007, pp. 264–271.

[9] Arthur Gretton, Karsten M. Borgwardt, Malte Rasch, Prof. Bernhard Schölkopf, and Alex J. Smola, "A kernel method for the two-sample-problem," in *Advances in Neural Information Processing Systems 19*, pp. 513–520. MIT Press, 2007.

[10] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos, "MMD GAN: towards deeper understanding of moment matching network," in *Advances in Neural Information Processing Systems*, 2017, pp. 2203–2213.

[11] Wei-Wei Lin, Man-Wai Mak, and Jen-Tzung Chien, "Multisource i-vectors domain adaptation using maximum mean discrepancy based autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 12, pp. 2412–2422, 2018.

[12] Xiaodong Cui, Jing Huang, and Jen-Tzung Chien, "Multi-view and multi-objective semi-supervised learning for HMM-based automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 1923–1935, 2012.

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[15] Youzhi Tu, Man-Wai Mak, and Jen-Tzung Chien, "Information maximized variational domain adversarial learning for speaker verification," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020, pp. 6449–6453.

[16] Jen-Tzung Chien and Chun-Lin Kuo, "Variational Bayesian GAN," in *Proc. of European Signal Processing Conference*, 2019, pp. 1454–1458.

[17] Jen-Tzung Chien and Kang-Ting Peng, "Neural adversarial learning for speaker recognition," *Computer Speech & Language*, vol. 58, pp. 422–440, 2019.

[18] Jen-Tzung Chien and Chun-Lin Kuo, "Bayesian adversarial learning for speaker recognition," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop*, 2019, pp. 381–388.

[19] Jen-Tzung Chien and Kang-Ting Peng, "Adversarial manifold learning for speaker recognition," in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop*, 2017, pp. 599–605.

[20] Youzhi Tu, Man-Wai Mak, and Jen-Tzung Chien, "Variational Domain Adversarial Learning for Speaker Verification," in *Proc. of Annual Conference of International Speech Communication Association*, 2019, pp. 4315–4319.

[21] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel, "The variational fair autoencoder," in *Proc. of International Conference on Learning Representations*, 2016.

[22] Diederik P. Kingma and Max Welling, "Auto-encoding variational Bayes," in *Proc. of International Conference on Learning Representations*, 2014.

[23] Jiayuan Huang, Arthur Gretton, Karsten M. Borgwardt, Bernhard Schölkopf, and Alexander J. Smola, "Correcting sample selection bias by unlabeled data," in *Advances in Neural Information Processing Systems*, 2006, pp. 601–608.

[24] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. of International Conference on Machine Learning*, 2015, pp. 97–105.

[25] Hung-Yu Chen and Jen-Tzung Chien, "Deep semi-supervised learning for domain adaptation," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, 2015, pp. 1–6.

[26] Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3581–3589.

[27] John Blitzer, Mark Dredze, and Fernando Pereira, "Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification," in *Proc. of Annual Meeting of the Association of Computational Linguistics*, 2007, vol. 7, pp. 440–447.

[28] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. of International Conference on Machine Learning*, 2011, pp. 513–520.

[29] Ji Zhao and Deyu Meng, "FastMMD: ensemble of circular discrepancy for efficient two-sample test," *Neural Computation*, vol. 27, no. 6, pp. 1345–1372, 2015.

[30] Diederik Kingma and Jimmy Ba, "Adam: a method for stochastic optimization," in *Proc. of International Conference on Learning Representations*, 2015.