2017 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 25–28, 2017, TOKYO, JAPAN

# ADVERSARIAL DOMAIN SEPARATION AND ADAPTATION

*Jen-Chieh Tsai* *Jen-Tzung Chien*

Department of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan

## ABSTRACT

Traditional domain adaptation methods attempted to learn the *shared* representation for distribution matching between source domain and target domain where the *individual* information in both domains was *not* characterized. Such a solution suffers from the mixing problem of individual information with the shared features which considerably constrains the performance for domain adaptation. To relax this constraint, it is crucial to extract both shared information and individual information. This study captures both information via a new domain separation network where the shared features are extracted and purified via separate modeling of individual information in both domains. In particular, a hybrid adversarial learning is incorporated in a separation network as well as an adaptation network where the associated discriminators are jointly trained for domain separation and adaptation according to the minmax optimization over separation loss and domain discrepancy, respectively. Experiments on different tasks show the merit of using the proposed adversarial domain separation and adaptation.

***Index Terms***— Deep learning, domain adaptation, latent features, adversarial learning, pattern classification

## 1. INTRODUCTION

Rapid development of deep learning has dramatically improved system performance in different classification and regression tasks including image recognition, speech recognition, source separation, text categorization, machine translation, etc. Basically, training a deep model in presence of large number of labeled samples can achieve state-of-the-art performance. But, in real-world applications, it is expensive and time-consuming to collect a large amount of labeled data. Besides, the training data from source domain behave differently in their distributions from the test data in target domain. System performance of a deep model is bounded due to the limited amount of labeled data and the mismatch between source and target domains. To deal with this issue, a branch of transfer learning, called domain adaptation, aims at extracting the domain invariant features from labeled data in source domain and unlabeled data in target domain. This method learns for a different but related target data distribution from a source data distribution which is beneficial for the objective of a new task.

In general, domain adaptation is performed to pursue distribution matching between two domains based on the corresponding latent features. Such a similarity matching was performed by minimizing the maximum mean discrepancy (MMD) between latent features in two domains [1]. In addition to MMD criterion, the optimization for distribution matching based on $\mathcal{H}$-divergence [2] and Jensen-Shannon divergence was developed for domain adaptation where deep models were involved to capture the complicated relation between data distributions of source and target domains. The feature extraction, pattern classification and data reconstruction were jointly optimized to build a deep model for domain adaptation [3]. In [4], the residual transfer network was proposed to learn residual function for adaptation from source classifier to target classifier according to MMD criterion. The transferable features were learned for image recognition by using the deep adaptation network based on convolutional neural network [4]. In [5], a variational fair autoencoder (VFAE) was proposed to estimate the invariant features for domain adaptation where the variational autoencoder was performed by incorporating MMD as a regularization term. This study presents a hybrid adversarial learning for joint optimization of a domain separation network and a domain adaptation network. A sophisticated representation of latent features for individual domains and shared classes is learned to build a classification network from labeled data in source domain and unlabeled data in target domain.

## 2. RELATED WORKS

In domain adaptation, training samples are collected from source domain $s$ and target domain $t$. Let $\{X_s, Y_s\} = \{\mathbf{x}_{sn}, \mathbf{y}_{sn}\}_{n=1}^{N_s}$ denote $N_s$ labeled examples in source domain where $\mathbf{x}_{sn}$ denotes the $n^{\text{th}}$ training vector and $\mathbf{y}_{sn}$ reflects its class label. For a task classifier with $K$ classes, $\mathbf{y}_{sn}$ is encoded as a vector by using 1-of-$K$ coding scheme. In addition, we have $N_t$ unlabeled examples $X_t = \{\mathbf{x}_{tn}\}_{n=1}^{N_t}$ from target domain where the label information $Y_t$ is missing. Basically, domains $s$ and $t$ 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 branch of transfer learning where the

978-1-5090-6341-3/17/$31.00 ©2017 IEEE

marginal distributions $p(X_s)$ and $p(X_t)$ are different and the conditional distributions of finding labels from data in two domains $p(Y_s|X_s)$ and $p(Y_t|X_t)$ are assumed to be identical. Domain adaptation is seen as a semi-supervised learning problem [6] for joint representation of data in source domain as well as target domain. The adaptation methods based on domain adversarial network (DAN) [2] and domain separation network (DSN) [7] provide insights for the proposed adversarial domain separation and adaptation.

### 2.1. Domain adversarial network

DAN [2] was proposed to extract the invariant features between source and target domains by using the adversarial learning, which was originally designed for construction of a generative model, known as the generative adversarial network (GAN) [8]. DAN ran the adversarial process for domain adaptation according to a measure of disparity between distributions in two domains through a deep discriminator model. Different from GAN matching the distributions of real data and synthesized data, the adverarial learning in DAN was performed to match the *latent features* in source domain and target domain. The architecture of DAN contained a feature extractor, a domain discriminator and a task classifier. The task classifier was trained to predict task-specific class label of an input while the domain discriminator was estimated to predict its domain label. A shared encoder or feature extractor was incorporated to generate the latent features which were *helpful* to predict class label but *harmful* to detect domain label. A hybrid optimization objective was formed by a classification loss and a domain regularizer due to the domain mismatch measured by $\mathcal{H}$-divergence. Therefore, DAN minimized this hybrid objective with respect to the parameters of feature extractor and task classifier and simultaneously maximized this objective with respect to the parameters of domain discriminator. The minimax optimization was realized by using gradient reversal layer which acted as an identity transformation in forward propagation. This layer took the gradient from subsequent layer and changed its sign before backpropagation to the preceding layer. In general, the latent features extracted by DAN were *shared* for both source and target domains. Similar to DAN, some other methods in [4, 9] were proposed by finding the shared features.

### 2.2. Domain separation network

In [7], DSN introduced the *individual* or private features of source domain and target domain in domain adaptation. There were “low-level” and “high-level” differences in the distributions of image data in source and target domains. Low-level difference reflected the domain-specific characteristics due to noise, resolution, illumination or color while high-level difference meant the number of classes or types of objects. DSN assumed that source and target domains differed from low-level image statistics but had the same label space or behaved with similar distributions on high-level statistics. Due to these considerations, the integrated objective in DSN consisted of four loss functions. The first loss function was the *cross-entropy* error function for the task classifier. The second loss function was the *reconstruction* loss for input samples which was obtained by decoding the shared and individual features and was calculated by a scale-invariant mean squared error term. The third loss function was the *similarity* loss which was implemented by imposing the shared features to be irrelevant to domain label through the way of using MMD loss [4] or DAN loss [2]. The fourth loss function was the *difference* loss between shared features and individual features which was measured by a soft subspace orthogonality constraint by using all samples in two domains. DSN was trained by jointly optimizing these four loss functions. Although DSN improves DAN by separately modeling the shared features and individual features, there are still twofold weaknesses by using DSN. First, the separation loss or difference loss is intuitive but too simple. Second, only the shared features are learned in domain adaptation. The individual features are only used for reconstruction but *not* learned for domain adaptation.

## 3. ADVERSARIAL SEPARATION AND ADAPTATION

This paper deals with these limitations by using a joint framework for domain separation and adaptation. A hybrid adversarial learning is proposed to extract informative features for class labels and domain labels from labeled data in source domain and unlabeled data in target domain.

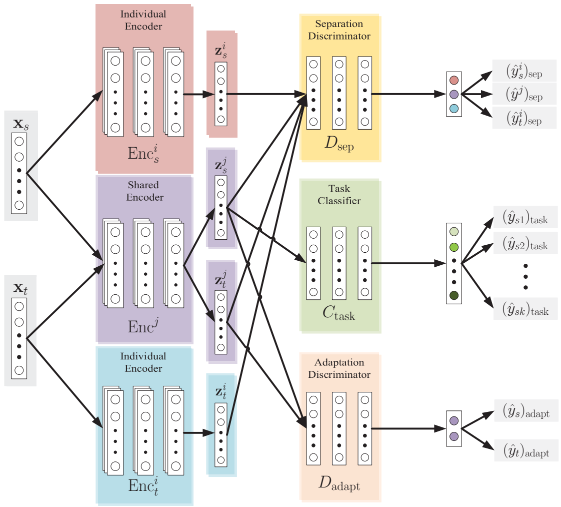


**Fig. 1**. A neural classification model based on adversarial domain separation and adaptation network.

### 3.1. System architecture

Figure 1 illustrates a neural classification model for domain adaptation based on the adversarial domain separation and adaptation (ADSA) network. $\mathbf{x}_s$ and $\mathbf{x}_t$ denote the input samples from source and target domains, respectively. For simplicity, we ignore sample index $n$. There are six neural networks $\{\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t, C_{\text{task}}, D_{\text{sep}}, D_{\text{adapt}}\}$ in ADSA model. These neural networks are categorized into two groups. The first group consists of three encoders including the shared encoder $\text{Enc}^j$ and the individual encoders $\text{Enc}^i_s$ and $\text{Enc}^i_t$ for source and target domains, respectively. The shared encoder is trained to extract joint or common features $\{\mathbf{z}^j_s, \mathbf{z}^j_t\}$ providing helpful information for task classification in both domains $\{s, t\}$. The individual encoder is trained to extract unique features $\{\mathbf{z}^i_s, \mathbf{z}^i_t\}$ reflecting individual source and target domains.

The second group contains three discriminators including the task classifier $C_{\text{task}}$, the adaptation discriminator $D_{\text{adapt}}$ and the separation discriminator $D_{\text{sep}}$. The task classifier predicts the class labels by using the shared features $\mathbf{z}^j_s$ in source domain encoded by shared encoder. Outputs of task classifier correspond to the class posteriors $\{(\hat{y}_{sk})_{\text{task}}\}_{k=1}^K$ where $(\hat{y}_{sk})_{\text{task}} = p(k|\mathbf{z}^j_s)$. The adaptation discriminator is adopted to find the measure of disparity for features between source domain $\mathbf{z}^j_s$ and target domain $\mathbf{z}^j_t$ encoded by shared encoder. Outputs of adaptation discriminator are binary for posterior probabilities of two domains $\{(\hat{y}_s)_{\text{adapt}}, (\hat{y}_t)_{\text{adapt}}\}$. Different from adaptation discriminator, the separation discriminator uses the shared features $\{\mathbf{z}^j_s, \mathbf{z}^j_t\}$ and the individual features $\{\mathbf{z}^i_s, \mathbf{z}^i_t\}$ as the inputs to produce three posterior outputs for shared features $(\hat{y}^j)_{\text{sep}}$ and individual features of source domain and target domain $\{(\hat{y}^i_s)_{\text{sep}}, (\hat{y}^i_t)_{\text{sep}}\}$. Importantly, separation network is incorporated to extract four sets of salient features which are separate mutually in accordance with two aspects. One is to separate between shared features and individual features while the other is to separate individual features between source domain and target domain. The resulting ADSA network tackles the weaknesses of DSN in [7] by introducing the sophisticated separation network for finding well-separated features $\{\mathbf{z}^j_s, \mathbf{z}^j_t, \mathbf{z}^i_s, \mathbf{z}^i_t\}$ by using adversarial learning which will be mentioned in Section 3.3. Also, the individual features $\{\mathbf{z}^i_s, \mathbf{z}^i_t\}$ are truly learned for domain adaptation based on the separation and adaptation networks.

### 3.2. Optimization procedure

ADSA network can be decomposed into three components: the *task classifier* for class label prediction, the *encoders* for shared features and individual features, and the *domain discriminators* for adaptation and separation. ADSA jointly optimizes an integrated loss function with respect to the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_{\text{task}}, \boldsymbol{\theta}_{\text{adapt}}, \boldsymbol{\theta}_{\text{sep}}\}$ where $\boldsymbol{\theta}_{\text{task}} = \{\boldsymbol{\theta}_{\text{Enc}^j}, \boldsymbol{\theta}_{C_{\text{task}}}\}$, $\boldsymbol{\theta}_{\text{adapt}} = \{\boldsymbol{\theta}_{\text{Enc}^j}, \boldsymbol{\theta}_{D_{\text{adapt}}}\}$ and $\boldsymbol{\theta}_{\text{sep}} = \{\boldsymbol{\theta}_{\text{Enc}^j}, \boldsymbol{\theta}_{\text{Enc}^i_s}, \boldsymbol{\theta}_{\text{Enc}^i_t}, \boldsymbol{\theta}_{D_{\text{sep}}}\}$ are fully-connected weight parameters

$$\begin{aligned} \min_{\{\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t, C_{\text{task}}, D_{\text{sep}}\}} \max_{D_{\text{adapt}}} \mathcal{L}_{\text{task}}(\text{Enc}^j, C_{\text{task}}) + \\ \mathcal{L}_{\text{adapt}}(\text{Enc}^j, D_{\text{adapt}}) + \mathcal{L}_{\text{sep}}(\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t, D_{\text{sep}}). \end{aligned} \tag{1}$$

For simplicity, we express the objective in terms of six neural networks instead of their parameters. Notably, similar to GAN, ADSA is running a minimax objective function which is minimized with respect to $\{\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t, C_{\text{task}}, D_{\text{sep}}\}$ but maximized with respect to $D_{\text{adapt}}$. There are three terms in this objective. First, we minimize the *classification loss* for class labels in task classifier which is calculated as a cross-entropy error function over training samples in source domain

$$\mathcal{L}_{\text{task}}(\text{Enc}^j, C_{\text{task}}) = -\sum_{n=1}^{N_s}\sum_{k=1}^{K} y_{snk} \log(\hat{y}_{snk})_{\text{task}} \tag{2}$$

where $\mathbf{y}_{sn} = \{y_{snk}\}_{k=1}^K$ is the class label of of a sample $n$ in source domain $s$ and $(\hat{\mathbf{y}}_{sn})_{\text{task}} = \{(\hat{y}_{snk})_{\text{task}}\}_{k=1}^K$ is the class posteriors which are calculated by using the parameters of shared encoder $\text{Enc}^j$ and task classifier $C_{\text{task}}$. Second, the parameters of adaptation discriminator $D_{\text{adapt}}$ and shared encoder $\text{Enc}^j$ are estimated according to adversarial learning based on a minimax objective using the *adaptation loss*

$$\min_{\text{Enc}^j} \max_{D_{\text{adapt}}} \Big\{ \mathcal{L}_{\text{adapt}}(\text{Enc}^j, D_{\text{adapt}}) = \mathbb{E}_{\mathbf{x}}[\log D_{\text{adapt}}(\text{Enc}^j(\mathbf{x}_s))] + \mathbb{E}_{\mathbf{x}}[\log(1 - D_{\text{adapt}}(\text{Enc}^j(\mathbf{x}_t)))] \Big\} \tag{3}$$

where two terms in right-hand-side (RHS) indicate the *negative cross-entropy* errors of a binary classifier. This objective is different from that of GAN which was designed for unsupervised learning but similar to that of DAN which was developed for domain adaptation [2]. ADSA encourages the distribution matching of shared features $\{\mathbf{z}^j_s, \mathbf{z}^j_t\}$ in two domains. Third, we minimize the *separation loss* to enhance the discrimination of latent features $\{\mathbf{z}^j_s, \mathbf{z}^j_t, \mathbf{z}^i_s, \mathbf{z}^i_t\}$ encoded by $\{\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t\}$ via a separation discriminator $D_{\text{sep}}$ based on the cross-entropy loss function

$$\mathcal{L}_{\text{sep}}(\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t, D_{\text{sep}}) = -\sum_{n=1}^{2(N_s+N_t)} \sum_{k=1}^{3} (y_{nk})_{\text{sep}} \log(\hat{y}_{nk})_{\text{sep}} \tag{4}$$

where $(\mathbf{y}_n)_{\text{sep}} = \{(y_{nk})_{\text{sep}}\}_{k=1}^3$ denotes the label vector for the groups of shared features and individual features for domains $s$ and $t$ using 1-of-3 coding scheme and $(\hat{\mathbf{y}}_n)_{\text{sep}} = \{(\hat{y}^j_n)_{\text{sep}}, (\hat{y}^i_{sn})_{\text{sep}}, (\hat{y}^i_{tn})_{\text{sep}}\}$ denotes the posterior outputs corresponding to three different groups. Number of samples in Eq. (4) is double because two encoders are used for each sample. Notably, treating the separation discriminator as a classification network is more intuitive than that in DSN driven by the orthogonality constraint. In addition to minimizing separation loss based on cross-entropy error function, we further introduce a new solution based on adversarial learning as described in what follows.

**Algorithm 1** ADSA implementation procedure

```
Initialize parameters $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_{D_\text{adapt}}, \boldsymbol{\theta}_{D_\text{sep}}, \boldsymbol{\theta}_{C_\text{task}}, \boldsymbol{\theta}_{\text{Enc}^j}, \boldsymbol{\theta}_{\text{Enc}^i_s}, \boldsymbol{\theta}_{\text{Enc}^i_t}\}$
for all samples $\{\mathbf{x}_s, \mathbf{x}_t\}$ do
    while $D_\text{adapt}$ not converged do
        $\boldsymbol{\theta}_{D_\text{adapt}} \to \boldsymbol{\theta}_{D_\text{adapt}} - \eta\nabla\mathcal{L}_\text{adapt}(\text{Enc}^j, D_\text{adapt})$
    end while
    while $\text{Enc}^j$ not converged do
        $\boldsymbol{\theta}_{\text{Enc}^j} \to \boldsymbol{\theta}_{\text{Enc}^j} + \eta\nabla\mathcal{L}_\text{adapt}(\text{Enc}^j, D_\text{adapt})$
    end while
    while $D_\text{sep}$ not converged do
        $\boldsymbol{\theta}_{D_\text{sep}} \to \boldsymbol{\theta}_{D_\text{sep}} + \eta\nabla\mathcal{L}_\text{sep}(\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t, D_\text{sep})$
    end while
    while $\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t$ not converged do
        $\boldsymbol{\theta}_{\text{Enc}^j} \to \boldsymbol{\theta}_{\text{Enc}^j} - \eta\nabla\mathcal{L}_\text{sep}(\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t, D_\text{sep})$
        $\boldsymbol{\theta}_{\text{Enc}^i_s} \to \boldsymbol{\theta}_{\text{Enc}^i_s} - \eta\nabla\mathcal{L}_\text{sep}(\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t, D_\text{sep})$
        $\boldsymbol{\theta}_{\text{Enc}^i_t} \to \boldsymbol{\theta}_{\text{Enc}^i_t} - \eta\nabla\mathcal{L}_\text{sep}(\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t, D_\text{sep})$
    end while
    $\boldsymbol{\Theta} \to \boldsymbol{\Theta} - \eta\nabla[\mathcal{L}_\text{task} - \mathcal{L}_\text{adapt} + \mathcal{L}_\text{sep}]$
end for
```

### 3.3. Hybrid adversarial learning

ADSA presents a truly adversarial approach which is not only carried out for adaptation network but also for separation network. In adaptation network, the adaptation loss is minimized by estimating the shared encoder $\text{Enc}^j$ which produces the maximal confusion set of shared features between source domain $\mathbf{z}^j_s$ and target domain $\mathbf{z}^j_t$. This adversarial optimization is performed by finding the neural network parameters of adaptation discriminator $\boldsymbol{\theta}_{D_\text{adapt}}$ via minimization of adaptation loss $\mathcal{L}_\text{adapt}(\text{Enc}^j, D_\text{adapt})$ and estimating those of shared encoder $\boldsymbol{\theta}_{\text{Enc}^j}$ via maximization of adaptation loss. The adversarial optimization in adaptation network aims to train an adaptation discriminator which minimizes the maximal adaptation loss that the estimation of shared encoder is caused. Under the same framework of ADSA, the adversarial optimization is also performed to estimate the parameters of separation network including those of shared encoder $\boldsymbol{\theta}_{\text{Enc}^j}$, individual encoders $\{\boldsymbol{\theta}_{\text{Enc}^i_s}, \boldsymbol{\theta}_{\text{Enc}^i_t}\}$ and separation discriminator $\boldsymbol{\theta}_{D_\text{sep}}$. Different from adaptation network, the separation network is constructed by training a shared encoder which *minimizes* the *maximal* separation loss $\mathcal{L}_\text{sep}(\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t, D_\text{sep})$ that the estimation of separation discriminator is caused. Namely, the trained encoders are forced to generate *good* features which can be correctly predicted by a *poor* discriminator. We comparably deal with the following minimax optimization

$$\min_{\{\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t\}} \max_{D_\text{sep}} \Big\{ \mathcal{L}_\text{sep}(\text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t, D_\text{sep}) \\ = -\mathbb{E}_\mathbf{x}[\log D_\text{sep}(\text{Enc}^j(\mathbf{x}_s)) + \log D_\text{sep}(\text{Enc}^j(\mathbf{x}_t))] \\ -\mathbb{E}_\mathbf{x}[\log D_\text{sep}(\text{Enc}^i_s(\mathbf{x}_s))] - \mathbb{E}_\mathbf{x}[\log D_\text{sep}(\text{Enc}^i_t(\mathbf{x}_t))] \Big\}. \tag{5}$$

In Eq. (5), we consider a *minimax* optimization which is consistent with that in adaptation network in Eq. (3). But, the objective has been changed to the *cross-entropy error* function. Three terms in RHS are calculated from outputs of a 3-class classifier. Algorithm 1 illustrates the implementation of ADSA domain adaptation by using hybrid adversarial learning. $\eta$ means the learning rate. Notably, after updating the parameters of $\{D_\text{adapt}, D_\text{sep}, \text{Enc}^j, \text{Enc}^i_s, \text{Enc}^i_t\}$, ADSA conducts an overall updating for all parameters including those for task classifier $C_\text{task}$ via $\boldsymbol{\Theta} \to \boldsymbol{\Theta} - \eta\nabla[\mathcal{L}_\text{task} - \mathcal{L}_\text{adapt} + \mathcal{L}_\text{sep}]$ where equal weights are adopted for three loss functions.

## 4. EXPERIMENTS

The proposed ADSA was examined for domain adaptation by using two tasks. The optimization using stochastic gradient descent (SGD) with momentum was applied.

### 4.1. Experimental setup

The first task was conducted for sentiment classification by using multi-domain sentiment dataset which consisted of Amazon product reviews on four product types or domains (books, DVDs, electronics, kitchen appliances). There were 1000 positive reviews (higher than 3 stars) and 1000 negative reviews (3 stars or lower) on each product domain, which were used as training data for binary classification. There were a number of test reviews in a range between 3000 and 5000 on each product domain. Each text document was represented 5K features which were extracted by using scikit-learning (http://scikit-learn.org/). These features contained bag-of-words unigrams and bigrams with tf-idf reweighting using a dictionary of 5K frequent words. In this task, ADSA was implemented by using the number of fully-connected layers to be three, three, three, two, three and two for neural network components $\text{Enc}^j$, $\text{Enc}^i_s$, $\text{Enc}^i_t$, $C_\text{task}$, $D_\text{adapt}$ and $D_\text{sep}$, respectively. In adversarial learning, we sequentially train the adaptation discriminator, the shared encoder, the separation discriminator and then all encoders by using 40, 4, 2 and 2 minibatches, respectively. 10K learning epochs were run.

The second task was evaluated for handwritten digit recognition by using three datasets or domains which are MNIST, MNIST-M and USPS. MNIST-M dataset was a distorted MNIST dataset where the digit and background of an MNIST image were treated as a binary mask which was cropped and then converted by using different background images randomly sampled from the Berkeley segmentation dataset [2]. For the adaptation between MNIST and MNIST-M, We consistently sampled 1000 examples from totally 59K training examples as validation data. There were 10K test examples. Following the adaptation between MNIST and USPS in [10], we sampled 2000 and 1800 examples from MNIST and USPS for training and 2000 and 2007 examples for testing, respectively. ADSA was implemented by using two convolution layers and two pooling layers for all encoders, three fully-connected layers for task classifier, two fully-connected layers for adaptation discriminator and two fully-connected layers for separation discriminator. We sequentially train the adaptation discriminator, the shared encoder, the separation discriminator and then all encoders by using 20, 5, 4 and 10 minibatches, respectively. 40K learning epochs were run.

For comparison, the baseline systems of using deep neural network (DNN) for the first task and convolutional neural network (CNN) for the second task were implemented. The classification results of VFAE [5], DAN [2] and DSN [7] were carried out. There were two ADSA realizations which were characterized by the optimization strategy of separation network. ADSA-I adopted the *single-pass* minimization over the separation loss measured by three encoders and one separation discriminator. ADSA-II applied the *minimax* optimization in construction of separation network where the separation loss was minimized with respect to separation discriminator but maximized with respect to three encoders. In what follows, we examine how these two optimization strategies affect the classification loss and clustering performance. The clustering was evaluated in terms of Davies Bouldin index (DBI) [11]. The smaller the DBI, the better the clustering.

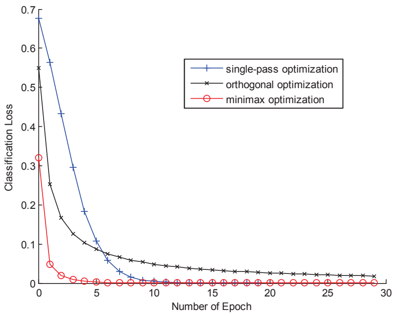


**Fig. 2**. Comparison of learning curves of using different optimization strategies. Amazon reviews are evaluated.

### 4.2. Evaluation for optimization strategies

First of all, we evaluate the learning behavior in ADSA-I and ADSA-II by comparing the learning curves of SGD optimization based on the minimization and minimax over the cross-entropy error function shown in Eqs. (4) and (5), respectively. For simplicity, this evaluation is only performed for the classification network consisting of a shared encoder $\text{Enc}^j$ and a task classifier $C_{\text{task}}$. Three fully-connected layers are built in a shared encoder with the number of neurons being 5000, 500 and 100. The optimization with orthogonality constraint on the features between classes [7] is also evaluated. As compared in Figure 2, minimax optimization converges faster with lower classification loss (or cross-entropy error function) than optimizations using orthogonal constraint and single-pass minimization. *Adversarial* learning over encoder and classifier does obtain better representation than traditional *single-pass* learning over both encoder and classifier.

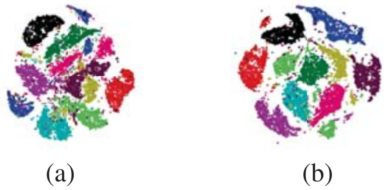


**Fig. 3**. Visualization of shared features $\mathbf{z}_s^j$ of MNIST digits estimated by (a) single-pass and (b) minimax optimizations.

Figure 3 compares the visualization of latent features $\mathbf{z}_s^j$ of MNIST digits extracted from the convolutional classification network consisting of a convolutional encoder and a task classifier where minimization and minimax optimizations over classification loss are performed. Colors indicate individual digits. The features encoded in the last pooling layer are visualized by applying the $t$-distributed stochastic neighbor embedding. Again, the adversarial learning using minimax optimization performs better in terms of DBI and obtains more separable convolutional features than that using single-pass optimization. DBI is measured by 86 and 79 by using single-pass and adversarial optimizations, respectively.

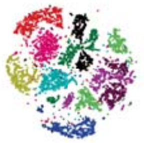
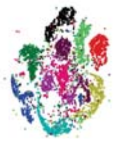
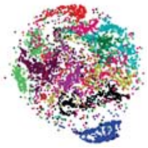
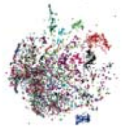
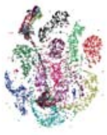
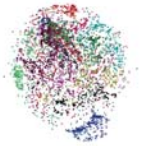
| | Baseline | DAN | ADSA |
|---|---|---|---|
| $\mathbf{z}_s^j$ |  |  |  |
| $\mathbf{z}_t^j$ |  |  |  |

**Table 1**. Visualization of shared features of digits in source domain $\mathbf{z}_s^j$ (MNIST) and target domain $\mathbf{z}_t^j$ (MNIST-M).

### 4.3. Experimental results

To evaluate the performance of domain adaptation, we first demonstrate the visualization of shared features of source domain $\mathbf{z}_s^j$ and target domain $\mathbf{z}_t^j$ learned by baseline, DAN and ADSA as shown in Table 1. Baseline means the CNN without domain adaptation. Obviously, the distributions of ten digits in source domain (MNIST) shape differently from those in target domain (MNIST-M) in baseline system. In general, DAN and ADSA obtain the shared features in MNIST which are matching with those in MNIST-M. If we look closely at the distributions of different digits, the digit in black color using DAN shapes differently between MNIST and MNIST-M.

Such a case is unseen by using ADSA. The distribution of different digits using ADSA is concentrated due to the effect of hybrid minimax optimization.

Tables 2 and 3 report the accuracies of sentiment classification and handwritten digit recognition under different cases of domain adaptation, respectively. In sentiment classification, DAN, DSN and ADSA consistently performs better than baseline DNN and VFAE. ADSA achieves higher accuracies than DAN and DSN in most cases. In handwritten digit recognition, we further compare the accuracies of using ADSA I and ADSA II with those of baseline CNN, DAN and DSN and find that ADSA II consistently outperforms ADSA I. The importance of adversarial learning in domain adaptation is confirmed. The proposed ADSA attains higher accuracies than other methods in most cases of domain adaptation. Source code of ADSA is accessible at https://github.com/NCTUMLlab/Jen-Chieh-Tsai-adsa.

| | Baseline | VFAE | DAN | DSN | ADSA |
|---|---|---|---|---|---|
| B → D | 77.2 | 77.6 | 78.4 | 78.2 | **80.2** |
| B → E | 70.3 | 69.0 | 73.3 | 71.8 | **78.3** |
| B → K | 72.8 | 70.3 | 77.9 | 76.6 | **81.3** |
| D → B | 74.2 | 71.4 | 72.3 | 73.6 | **76.4** |
| D → E | 71.8 | 71.2 | 75.4 | 74.8 | **77.7** |
| D → K | 75.6 | 73.5 | 78.3 | 79.7 | **80.1** |
| E → B | 70.9 | 68.8 | 71.1 | 69.7 | **72.6** |
| E → D | 67.9 | 69.7 | 73.8 | 74.1 | **74.5** |
| E → K | 73.0 | 81.5 | 85.4 | **86.4** | 85.5 |
| K → B | 66.9 | 66.8 | 70.9 | 69.4 | **72.5** |
| K → D | 68.0 | 68.4 | 74.0 | 71.3 | **75.8** |
| K → E | 82.5 | 82.4 | **84.3** | 83.0 | 83.6 |
| Average | *73.4* | *72.5* | *76.3* | *76.1* | *78.2* |

**Table 2**. Classification accuracies (%) for adaptation among different domains (B: Books, D: DVDs, E: Electronics, K: Kitchen appliances).

| | Baseline | DAN | DSN | ADSA I | ADSA II |
|---|---|---|---|---|---|
| M → MM | 56.6 | 76.6 | 83.2 | 81.2 | **84.1** |
| M → U | 59 | 63.4 | 69.6 | 76.1 | **80.8** |

**Table 3**. Classification accuracies (%) for adaptation among M: MNIST, MM: MNIST-M and U: USPS.

## 5. CONCLUSIONS

This paper presented a hybrid adversarial learning for adaptation discriminator and separation discriminator which could sufficiently distinguish the shared and individual features of source domain and target domain estimated by the shared and individual encoders, respectively. We jointly performed domain separation and adaptation to identify informative features for an improved classification performance. Experiments on two tasks demonstrated the meaningfulness of minimax optimization and estimated features for sentiment classification and image recognition.

## 6. REFERENCES

[1] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, pp. 723–773, 2012.

[2] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, 2016.

[3] H.-Y. Chen and J.-T. Chien, "Deep semi-supervised learning for domain adaptation," in *Proc. of International Workshop on Machine Learning for Signal Processing*, 2015, pp. 1–6.

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

[5] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, "The variational fair autoencoder," in *Proc. of International Conf. on Learning Representations*, 2016.

[6] X. Cui, J. Huang, and J.-T. 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.

[7] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 343–351.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[9] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," *arXiv preprint arXiv:1702.05464*, 2017.

[10] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. of IEEE International Conference on Computer Vision*, 2013, pp. 2200–2207.

[11] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE T-PAMI*, pp. 224–227, 1979.